Foreign Exchange UI

Semesterarbeit

Abteilung Informatik

Hochschule für Technik Rapperswil

HS09

Autoren	Daniel Häfliger
	Dominik Süsstrunk
Betreuer	Prof. Dr. Markus Stolze
Projektpartner	Dr. Beat Liver (Credit Suisse, Zürich)
Gegenleser	Prof. Dr. Josef Joller

Inhaltsverzeichnis

Aufgabenstellung Foreign Exchange UI

1 Auftraggeber und Betreuer

Diese Studienarbeit findet in Zusammenarbeit mit der Credit Swiss statt.

1.1 Ansprechpartner & Auftraggeber

- Dr. Beat Liver, Credit Suisse, IT, Treasury Flow Products, KSTC 11, beat.liver@credit-suisse.com

1.2 Betreuer HSR

- Prof. Dr. Markus Stolze, Institut für Software mstolze@hsr.ch

2 Ausgangslage

Gegeben ist eine White-labelled Forex Trading Internet-Applikation, welche aus Java-Applets und HTML-Masken besteht.[1]

Die Handelsmaske für Devisen unterstützt folgende Preisstellungsmethoden:

- Bei der Request-for-Quote Preisstellung werden die Geschäftsdetails erfasst und ein Wechselkurs wird angefragt (vlg. Demoversion 1). Dies ist somit ein Pull-Modell.
- Bei der Preisstellung mittels streamed handelbaren Kursen erfasst der Benutzer die Geschäftsdetails — wenigstens das Währungspaar — und subskribiert damit die entsprechenden Kurse vom Server. Dies ist somit ein Push-Modell.

3 Ziele und Aufgabenstellung

Entwurf und Implementation einer Geschäftserfassungsmaske auf neuerer Graphical User-Interface Technologie, wobei folgende Ziele vorgegeben sind:

- das Konzept soll grundsätzlich für die gesamte Produktepalette verwendbar sein muss, d.h. von den einfachen Kassageschäften bis zu den komplizierten Limitierten Devisenaufträgen, wie One-Cancels-the-Other (OCO). Die Wahl der Graphical User-Interface Technologie der neueren Generation soll begründet werden.
- Die Implementation soll die Erfassung von Devisenkassageschäften erlauben und dies sowohl im Pull- als auch Push-Mode, wobei die beiden Modi typischerweise die Benutzerklassen Privatanwender und professionelle Anwender abdecken. Man beachte, dass die serverseitige Präsentationslogik und Geschäftslogik (in J2EE-Terminologie währen dies dann Servlet und Entrerprise Java Beans) möglichst realitätsnahe bzw. simuliert werden sollen.

Die Entwurfskriterien sind Benutzbarkeit (Usability), hohe Leistung insbesondere geringe Latenzzeiten, Antwortzeiten, etc., Sicherheit, kurze Entwicklungszyklen, Wiederverwendung, Modularität und Wartbarkeit. Die Reihenfolge der Aufzählung ist die Gewichtung der Entwurfsziele. Evaluation wie gut die die Entwurfskriterien erreicht wurden.

[1] Siehe https://entry.credit-suisse.ch/csfs/p/cb/de/online/onl_uebersicht.jsp unter dem Menu Forex Trading > Demoversion.

Für diese Aufgabe kann die bestehende Anwendung benutzt werden und es stehen die entsprechende Dokumentation zur Verfügung. Eine Liste von Applikationen ist vorhanden, welche marktführende Elemente enthalten.
Im Weiteren können die entsprechenden Benutzer bzw. Benutzervertreter befragt werden und es können auch Benutzertests durchgeführt werden.
In einem ersten Schritt soll ein Testbed entwickelt werden welches die notwendigen Test Datafeeds zur Verfügung stellt.

4 Mögliche Erweiterungen

Erweiterung der Implementation für weitere Benutzerklassen, wie Semi-professionelle Anwender; weitere und komplizierte Produkte, wie Swaps; weitere Funktionen, wie Drucken, Import und Export.

5 Zur Durchführung

Mit den HSR-Betreuern finden in der Regel wöchentliche Besprechungen statt. Zusätzliche Besprechungen sind nach Bedarf durch die Studierenden zu veranlassen. Besprechungen mit dem Auftraggeber werden nach Bedarf durchgeführt.
Alle Besprechungen sind von den Studenten mit einer Traktandenliste vorzubereiten und die Ergebnisse in einem Protokoll zu dokumentieren, das den Betreuern, dem technischen Berater und dem Auftraggeber per E-Mail zugestellt wird.
Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen gemäss Projektplan sind einzelne Arbeitsresultate in vorläufigen Versionen abzugeben. Über die abgegebenen Arbeitsresultate erhalten die Studierenden ein vorläufiges Feedback. Eine definitive Beurteilung erfolgt auf Grund der am Abgabetermin abgelieferten Dokumentation.

6 Dokumentation

Über diese Arbeit ist eine Dokumentation gemäss den Richtlinien der Abteilung Informatik zu verfassen. Die zu erstellenden Dokumente sind im Projektplan festzuhalten. Alle Dokumente sind nachzuführen, d.h. sie sollten den Stand der Arbeit bei der Abgabe in konsistenter Form dokumentieren. Die Dokumentation ist vollständig auf CD/DVD in 5 Exemplaren abzugeben. Auf Wunsch ist für den Auftraggeber eine gedruckte Version zu erstellen. Zudem ist eine kurze Projektresultatdokumentation im Wiki von Prof. Stolze zu erstellen. Weiterhin erwünscht ist die Erstellung eines kurzen Videos oder einer Camtasia Demo sowie die Erstellung einer Kurzzusammenfassung (bis maximal 5 Seiten) welche die wichtigsten Resultate und Erkenntnisse der Arbeit zusammenfasst. Abteilung Informatik Herbstsemester 2009 Studienarbeit für Daniel Häfliger und Dominik Süsstrunk Foreign Exchange User Interface Seite 3/1 Markus Stolze Datei: SA-Häfliger-Süsstrunk-CS-FX.docx Ausgabe: 1.0 Letzte Änderung am: 14.09.09

7 Termine

Siehe auch Terminplan auf https://www.hsr.ch/Termine-Diplom-Bachelor-und.5142.0.html

14.09.2009	Beginn der Studienarbeit, Ausgabe der Aufgabenstellung durch die Betreuer
15.9.2009 8:00-9:00	Kick-off Sitzung mit Auftraggeber an HSR 6.112
18.12.2009	Abgabe Kurzbeschreibung an das Abteilungssekretariat mit folgendem Formular gemäss https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html
18.12.2009, 17:00	**Abgabe** des Berichtes an die Betreuer

8 Beurteilung

Eine erfolgreiche Studienarbeit erhält 8 ECTS-Punkten (1 ECTS Punkt entspricht einer Arbeitsleistung von ca. 25 bis 30 Stunden). Für die Modulbeschreibung der Studienarbeit siehe https://unterricht.hsr.ch/staticWeb/allModules/10938_M_SAI.html

Gesichtspunkt	Gewicht
1. Organisation, Durchführung	1/5
2. Berichte (Abstract, Mgmt Summary, techn. u. persönliche Berichte) sowie Gliederung, Darstellung, Sprache der gesamten Dokumentation	1/5
3. Inhalt*	3/5

*) Die Unterteilung und Gewichtung von 3. Inhalt wird im Laufe dieser Arbeit festgelegt.

Im Übrigen gelten die Bestimmungen der Abt. Informatik zur Durchführung von Studienarbeiten.
Rapperswil, den 14. September 2009



Prof. Dr. Markus Stolze
Institut für Software
Hochschule für Technik Rapperswil

Erklärung über die eigenständige Arbeit

Wir erklären hiermit,

- dass wir die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt haben, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben.

Dominik Süsstrunk, Rapperswil, den 17. Dezember 2009



Daniel Häfliger, Rapperswil, den 17. Dezember 2009



Abstract

1 Aufgabe

Forex Trading beschreibt das Handeln mit Währungen auf dem Foreign Exchange Market, dem grössten Finanzmarkt der Welt. Jedermann kann auf diesem Markt mit entsprechender Software (MetaTrader, FX Expert, etc.) handeln. Die Credit Suisse betreibt eine eigene Online Applikation (FX Expert), welche auf Java Applets basiert. Die Aufgabe besteht darin, die Machbarkeit einer Silverlight Applikation bezüglich Bedienung, Skalierbarkeit, Sicherheit und Wartbarkeit zu analysieren. Ziel ist es ein Testbed, welches die Bank simuliert, und einen Prototyp (Silverlight Client) zur Verfügung zu stellen.

2 Architektur

Der Webserver dient zugleich als Testbed und hosted die Client-Applikation die beim Aufruf der Website des Servers heruntergeladen und gestartet wird. Der Server stellt einen WCF-Service zur Verfügung. Dieser dient zur Kommunikation zwischen Server und Client, verwaltet die Benutzer und führt die Bankprozesse aus. Die Datenbank wird vom Server über Linq to SQL angesprochen. Die ganze Applikation ist in C# geschrieben.

3 Ergebnisse

Der Prototyp zeigt, dass sich Silverlight schon sehr gut für professionelle Anwendungen einsetzen lässt. Eine einfache Bedienung wurde durch ein übersichtliches und einfaches Design mit selbsterklärenden Controls (durch Tooltips, Bilder, etc.) erreicht. Der Benutzer bekommt vom User Interface Inputs zu wichtigen Aktionen (durch Bilder, Animation, etc.). Das Trennen der verschiedenen Schichten durch das MVVM Pattern, WCF und Linq to SQL führt zu einer guten Wartbarkeit der Applikation. Die Skalierbarkeit von Silverlight ist stark von Animationen abhängig. Durch auslagern des Zeichnen der aufwändigen Animationen auf die GPU und einstellen einer tieferen Framerate wurde hier eine Leistungssteigerung erreicht. Aus Zeitmangel wurde die serverseitige Skalierung nicht vertieft betrachtet, was eine gute Aufgabe für Folgearbeiten wäre. Die Sicherheit der Applikation ist die grosse Aufgabe, welche noch stark verbessert werden muss, da sie nur sehr rudimentär implementiert wurde.

Management Summary

1 Ausgangslage

Forex Trading beschreibt das Handeln mit Währungen auf dem Foreign Exchange Market, dem grössten Finanzmarkt der Welt. Jedermann kann auf diesem Markt mit entsprechender Software (MetaTrader, FX Expert, etc.) handeln. Die Credit Suisse betreibt eine eigene Online Applikation (FX Expert), welche auf Java Applets basiert. Die Aufgabe besteht nun darin, die Machbarkeit einer Silverlight Applikation bezüglich Bedienung, Skalierbarkeit, Sicherheit und Wartbarkeit zu analysieren. Ziel ist es ein Testbed, welches die Bank simuliert, und einen Prototyp (Silverlight Client) zur Verfügung zu stellen.

2 Vorgehen

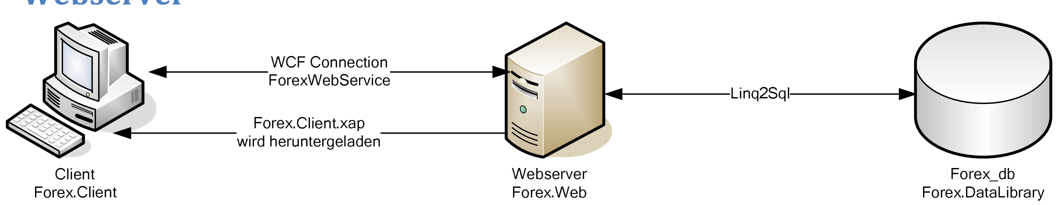
In einem ersten Schritt wurde ein Testbed entwickelt. Das Testbed erzeugt (zufällige, nicht realitätsgetreue) Devisenkurse, verwaltet die Benutzer und führt die Trades aus. Danach wurde der Prototyp, die Silverlight Applikation, entwickelt.

3 Technologien

3.1 Client

Der Client wurde mit Silverlight 3 entwickelt. Microsoft Silverlight ist ein Plug-In für Webbrowser, welches auf .NET basiert und die Ausführung von Rich Internet Applications ermöglicht und damit über die Möglichkeiten des klassischen HTML hinausgeht. Dem Anwender wird z.B. Drag and Drop, 3D-Effekte und Animationen ermöglicht. Zudem erfolgt meist eine schnellere Reaktion als bei klassischen HTML-basierten Anwendungen, da die Silverlight Applikationen auf dem lokalen Rechner ausgeführt werden und somit nicht auf die Reaktion eines Servers gewartet werden muss. Silverlight ist als proprietäres, programmierbares Plug-in für Windows und Apple Macintosh verfügbar. Für Linux wird von Novell mit Zustimmung und Unterstützung von Microsoft Moonlight angeboten. Im November 09 wurde die erste Beta zu Silverlight 4 veröffentlicht. Das User Interface wird in XAML beschrieben. Die Logik dahinter wird in C# definiert.

3.2 Webserver

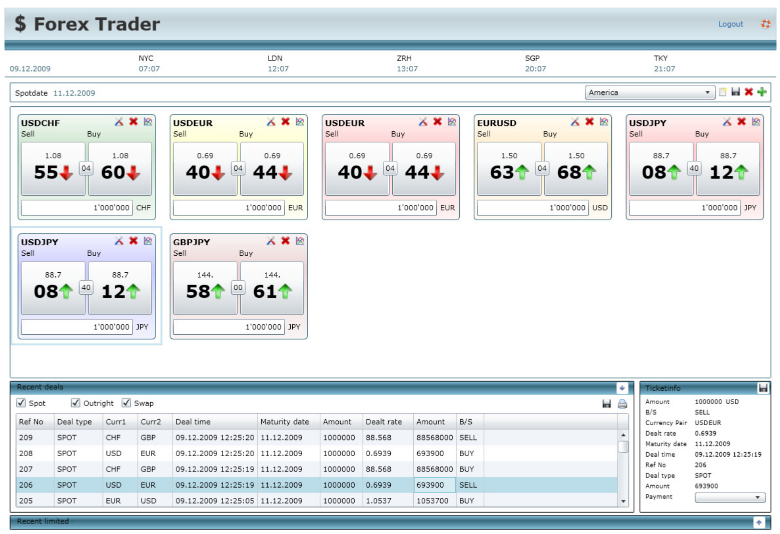


Der Webserver, welcher unter IIS6 läuft, wurde auch in C# geschrieben. Die Kommunikation zwischen dem Client und dem Server wird über einen WCF Service geregelt. WCF (Windows Communcation Foundation) stellt den Webservice zur Verfügung welcher für die Kommunikation zwischen Server und Client (Silverlight Applikation) gebraucht wird. Silverlight unterstützt hier nur das sogenannte „basicHttpBinding“, was dazu führt, dass der Service nicht Duplex verfügbar ist, sondern nur durch

Push vom Client her.
Die Datenbank wird vom Server über Linq to SQL angesprochen.
Der Webserver dient zugleich als Testbed und hosted die Client-Applikation die beim Aufruf der Website des Servers heruntergeladen und gestartet wird. Der Webservice kommuniziert mit dem Client und der Datenbank und verwaltet die Benutzer, platziert Trades und generiert die Rates.

4 Ergebnisse

Der Prototyp zeigt, dass sich Silverlight schon sehr gut für professionelle Anwendungen einsetzen lässt. Eine einfache Bedienung wurde durch ein übersichtliches und einfaches Design mit selbsterklärenden Controls (durch Tooltips, Bilder, etc) erreicht. Der Benutzer bekommt vom User Interface Inputs zu wichtigen Aktionen (durch Bilder, Animation, etc). Das Trennen der verschiedenen Schichten durch das MVVM Pattern, WCF und Linq to SQL führt zu einer guten Wartbarkeit der Applikation. Die Skalierbarkeit von Silverlight ist stark von Animationen abhängig. Durch auslagern des Zeichnen der aufwändigen Animationen auf die GPU und einstellen einer tieferen Framerate wurde hier eine Leistungssteigerung erreicht. Die Sicherheit der Applikation ist die grosse Aufgabe, welche noch stark verbessert werden muss, da sie nur sehr rudimentär implementiert wurde.



Ein Silverlight User Interface ist einfach zu gestalten, da durch Expression Blend ein professionelles Werkzeug für Designer zur Verfügung steht. Die Controls, welche von Silverlight zur Verfügung gestellt werden, haben auch schon einen modernen Default-Style, welchen wir meist nicht weiter verändert haben.
Durch Vergleiche von ähnlichen Applikation der verschiedenen RIA-Technologien (Java, Flash/Flex, Silverlight) ist uns aufgefallen, dass die Prozessorauslastung bei der Silverlight Applikation um einiges geringer ist als bei der Konkurrenz.
Was noch fehlt bei Silverlight 3 ist, dass die Trennung mittels des MVVM Patterns noch sehr mühsam zu Implementieren ist, da Silverlight nicht alle erforderlichen Klassen unterstützt.

5 Ausblick

Die Sicherheit bei der Applikation muss noch stark verbessert werden um einer Anwendung für eine Grossbank gerecht zu werden. Vor allem der WCF-Service, welcher nur über ein basicHttpBinding mit dem Server kommuniziert ist hier eine grosse Schwachstelle.
Weiter ist zu hoffen, dass mit dem offiziellen Release von Silverlight 4 auch die Unterstützung für das MVVM Pattern sowie die Handhabung der WCF Services verbessert wird.

Foreign Exchange UI

Technischer Bericht

1 Dokumentinformationen

1.1 Änderungsgeschichte

Datum	*Version*	*Änderung*	*Autor*
11.12.09	0.1	Dokument erstellt	ds
11.12.09	0.2	Dokument geschrieben	dh
12.12.09	1.0	Review	ds
15.12.09	1.1	Überarbeitung / Verbesserung	ds

2 Einleitung und Übersicht

Das Ziel war es, die Möglichkeiten von Silverlight anhand einer Applikation für den Devisenhandel zu erforschen. Wichtigste Ziele waren dabei, dass die Applikation modular aufgebaut wird, geringe Latenzen garantiert, kurze Entwicklungszeiten für Folgearbeiten haben und ein einfaches Warten möglich sein sollte. Das User Interface soll verbessert und intuitiv bedienbar sein und es soll modern aussehen.

Es sollen Währungspaare und deren aktuelle Kurse angezeigt werden, zu diesen Kursen sollten auch SPOT, SWAP sowie Outright Geschäfte vollzogen werden können. Limitierte Aufträge sollen erfasst und in einem Graphen dargestellt werden können.

3 Ergebnisse

3.1 Erreichte Ziele

- Modularer Aufbau
- Kurze Entwicklungszeiten
- User Interface
- Abonnieren von Währungspaaren
- FxGraph-Control mit grafischer Erfassung von limitierten Aufträgen

3.1.1 Erklärung

Der modulare Aufbau wurde durch strikte Trennung vom Code/User Interface und Einteilung in verschiedene Unterordner/Namespaces erreicht. Dies bewirkt auch die kurzen Entwicklungszeiten, da der Code in einer sauberen Struktur ist, findet man sich auch schnell darin zurecht.
Das User Interface wurde vor allem durch einige Paper Prototypen entwickelt und später anhand dieser implementiert.
Das Abonnieren von Währungspaaren funktioniert, leider abonniert man keinen Stream, da dies bei Silverlight 3 nicht über einen WCF Service funktioniert, jedoch funktioniert es, wenn man den Service nach jeder erhalten Antwort erneut anfragt (Pull-Mode).
Die grafische Erfassung von limitierten Aufträgen wird mittels Maus-Gesten gesteuert. Diese aktiviert man durch einen Klick, wenn man gleichzeitig die Control-Taste gedrückt hält. (Rechtsklick erst ab Silverlight 4 unterstützt)

3.2 Nicht erreichte Ziele

- Detaileingabe von SPOT/SWAP/Outright
- LimitOrder editieren
- Kerzengraph
- Moving Average in Graph
- Browsen in Graph

3.2.1 Ursachen

Die Ursachen der nicht erreichten Ziele sind praktisch alle durch Zeitmangel entstanden. Vorbereitet wäre das editieren von limitierten Aufträgen und das Erfassen der Detaileingaben von SPOT/SWAP/Outright.
Den Kerzengraph zu implementieren wäre nicht gerade simpel, da der Graph bis anhin nur mit Polylines gezeichnet wird, was bei einem Kerzengraph nicht funktioniert. Allerdings ist das Control so aufgebaut, dass man solche Änderungen problemlos einbetten könnte. Auch das Browsen der Daten im Graph wäre nicht gerade simpel, trotzdem auch ohne weiteres machbar.
Die History-Raten wahlweise als Moving Average zu laden und die Daten auch so darzustellen ist nicht sehr Zeitintensiv, man hätte beim aktualisieren der Raten, die älteren Raten nachberechnen müssen, die Daten sind aber nicht dafür vorbereitet und deswegen haben wir uns dagegen entschieden.

4 Schlussfolgerung

Gemäss Titel der Arbeit ging es vor allem um das User Interface, es wurde auch verhältnismässig viel Zeit darin investiert. Grundsätzlich wurde das User Interface bis auf einige Kleinigkeiten fertig designt, allerdings wurden die Funktionen teilweise nicht implementiert – die meisten nicht implementierten Funktionen wurden aber sauber vorbereitet und könnten ohne grossem Zeitaufwand noch fertiggestellt werden. Würde dieses Projekt nochmals gestartet, würden wir von Anfang an mehr Zeit für die Strukturierung der Code-Projekte verwenden, nur durch eine saubere Grundstruktur ist es möglich, eine saubere Software zu entwickeln.

Wir werden dieses Projekt in den nächsten Wochen bei der Credit Suisse vorstellen, was damit passiert können wir zu diesem Zeitpunkt noch nicht sagen, wir denken allerdings, dass unser Projekt eher als Forschungsarbeit denn als später zu verwendetes Projekt weiterleben wird. Es wäre aber durchaus möglich, dass einige unserer Designentscheidungen in einem allfälligen Projekt der Credit Suisse diskutiert werden.

Persönliche Berichte

1 Daniel Häfliger

1.1 Allgemein

Ich hatte schon immer ein Flair für Anwendungen im Internet, allerdings hatte ich immer nur relativ statische Seiten betreut. Silverlight war anfangs für mich eine grosse Horizonterweiterung, da ich oft mit .NET arbeite fand ich die Arbeit interessant. Überrascht hat mich das einfache Handling von Silverlight, damit hatte ich nicht gerechnet.

1.2 Ablauf des Projektes

Als die Themen für die Semesterarbeit zur Auswahl standen, fiel mir sofort dieses Projekt ins Auge. So war es nicht verwunderlich, dass ich mit sehr viel Elan in dieses Projekt gestartet bin. Anfangs war das Einarbeiten in die Bankenwelt sowie in Silverlight im Vordergrund. Beides sorgte ab und an für Verwirrung, doch je länger das Projekt lief, desto besser kam ich damit zu recht.
Das Paper-Prototyping hat mir sehr viel Spass gemacht, es ist ein netter Ausgleich zum sonstigen Programmieralltag.
Natürlich blieb unser Projekt auch nicht von Fehlern verschont. Der Versuch, sich mit dem Team Foundation Server zu versuchen misslang auf allen Ebenen, mit SVN war dann dieses Problem gelöst. Schade, hätte ich doch gerne die TFS-Welt besser kennengelernt.
Sonst war der Ablauf oft wie geplant. Differenzen werden immer zu finden sein, allerdings bewegt sich das Meiste in einer Marge, welche ich als akzeptabel erachte. Zum Schluss hat es dann leider nicht mehr gereicht, um alle Aufgabenbereiche komplett zu implementieren, alles in Allem ist es aber eine gute Übersicht.

1.3 Teamarbeit

Dominik Süsstrunk und ich bilden ein gutes Team, wir ergänzen uns prima. Wir konnten beide gegenseitig voneinander profitieren und haben für die Zukunft einiges gelernt. Das Arbeitsklima war stets von Respekt und Anstand geprägt und auch kleinere Differenzen konnten wir mit einer Diskussion klären, sodass beide zufrieden waren.

1.4 Fazit

Ich würde dieses Thema nochmals wählen und auch den Aufwand nochmals in Kauf nehmen. Das Projekt hat Spass gemacht und ich war stets motiviert. Mit dem jetzigen Wissensstand wäre es einiges einfacher, dieses Projekt aufzuziehen, viel ändern würde ich trotzdem nicht.
Alles in allem denke ich, dass uns diese Arbeit gut gelungen ist und für die Credit Suisse einen guten Einblick in die Silverlight Technologie gibt. Auch wenn es vielleicht zurzeit noch nicht so empfehlenswert ist, eine Finanzapplikation mit Silverlight zu programmieren, denke ich, dass in der Zukunft vermehrt solche RIA-Technologien eingesetzt werden.
Wird Silverlight 4 die Features welche noch zu beschränkt sind (wie z.B. WCF Service Bindings) erweitern, dann sehe ich dafür eine grosse Zukunft.

2 Dominik Süsstrunk

2.1 Allgemein

Silverlight hat mich in seinen Bann gezogen. Anfangs war ich noch skeptisch gegenüber Silverlight, da es abhängig von .NET ist und ich mir nicht sicher war, ob sich Silverlight jemals durchsetzen wird. Doch nach der Einarbeitung in das Thema wurde mir schnell bewusst, dass Silverlight in den meisten Dingen wohl schon weiter ist als vergleichbare Technologien wie Flash oder Java.

2.2 Ablauf des Projektes

Motiviert wie wir Jungen so sind, stürzten wir uns natürlich auf das Projekt und hatten viele Ideen für den Prototyp, die Versionierung und das Testbed.
Die grösste Zeitliche Fehlinvestition für mich war dann auch, dass wir die Versionierung mit einem Team Foundation Server bewerkstelligen wollten. Dies kostete mich ca. eine Woche. Die Änderung auf SVN dauerte dann gerade einmal noch 30 Minuten.
Mit der Zeit wurde mir auch bewusst, dass wir alle uns aufgetragenen Funktionen für den Prototyp und das Testbed wohl nicht erreichen würden, was dazu führte, dass wir relativ lange an der Implementation herum schraubten und so mit der Dokumentation nicht ganz im Zeitplan waren. Von den getätigten Stunden waren wir eigentlich die ganze Zeit im Soll, ausser auf die letzten zwei Wochen, welche doch noch einiges an Arbeit gebracht haben.

2.3 Teamarbeit

Die Zusammenarbeit mit Daniel Häfliger funktionierte meinem Erachten nach meist sehr gut. Wir arbeiteten fast nie am selben Teil der Applikation herum, was dazu führte, dass wir uns nicht allzu oft in die Quere gekommen sind. Auch die Dokumentation wurde klar aufgeteilt, sodass wir uns auch hier nicht gegenseitig behinderten. Die einzigen Probleme entstanden dann auch, wenn einer von uns Zwei nur sehr sporadisch seine geänderten Dateien eingecheckt hat. Dies führte ein bis zwei Mal zu kleineren Konflikten.

2.4 Fazit

Mit dem Endprodukt bin ich zufrieden. Natürlich hat es noch einige Dinge die noch nicht komplett abgearbeitet sind und man könnte hier natürlich noch unzählige Stunden investieren, doch für einen ersten Prototypen, um zu zeigen was mit Silverlight möglich ist, ist uns dies durchaus gelungen.
Das Projekt im Zusammenhang mit Silverlight und der ganzen Foreign Exchange Geschichte hat mir eigentlich gut gefallen und ich war auch stets motiviert, jedoch hatte ich mit der Zeit das Gefühl, dass eine Bankenapplikation für Silverlight-Beginner nicht gerade geeignet ist und ich mir somit ein wenig mehr Vorahnung in diesem Bereich gewünscht habe. Internettechnologien im Allgemeinen finde ich eine ganz spannende Sache. Ich hoffe auch in Zukunft mit diesen Technologien zu tun zu haben. Ein Vergleich zwischen Flex, Silverlight und anderen Technologien wäre sehr interessant.
Bei einem nächsten Projekt würde ich vor allem darauf achten, von Beginn weg nicht zu viele Ziele zu verfolgen, darauf die wenigen die übrig bleiben, sauber und komplett zu erfüllen. Auch würde ich die Softwareaufteilung bei Beginn genauer Aufteilung und sauber benennen.

Literaturverzeichnis

Dokument	*Autor*
SA-Häfliger-Süsstrunk-CS-FX Aufgabenstellung.pdf	ms
Projektantrag.pdf	Team
Internal Coding Guidelines http://blogs.msdn.com/brada/articles/361363.aspx	Brad Adams
Java Applet Version der bestehenden Anwendung https://entry.credit-suisse.ch/csfs/p/cb/de/online/onl_uebersicht.jsp	CS
Silverlight 3.0 http://www.silverlight.net	Microsoft
Anleitung zur bestehenden Java-Applet Version Manual_FX_Expert.pdf	CS
DbLINQ http://code.google.com/p/dblinq2007/	Google
Datenbank Provider für LINQ http://en.wikipedia.org/wiki/Language_Integrated_Query#Other_providers	WIKI
Orders, Dokument Credit-Suisse https://entry.credit-suisse.ch/csfs/p/cb/de/dev_zinsen/media/pdf/dzs_stoploss_de.pdf	CS
Brief- und Geldkurs Einträge in Wikipedia.org http://de.wikipedia.org/wiki/Briefkurs http://de.wikipedia.org/wiki/Geldkurs	WIKI
Termingeschäft, Dokument Credit-Suisse https://entry.credit-suisse.ch/csfs/p/cb/de/dev_zinsen/media/pdf/dzs_termingeschaeft_de.pdf	CS
SL Extensions http://www.slextensions.net/	SL Extensions
Silverlight 3 Toolkit July/October 2009 http://www.codeplex.com/Silverlight	Codeplex

Foreign Exchange UI

Projektplan

1 Dokumentinformationen

1.1 Änderungsgeschichte

Datum	Version	Änderung	Autor
18.09.09	0.1	Dokument erstellt, Projektorganisation/Management Abläufe definiert	Team
22.09.09	0.2	Infrastruktur und Qualitätsmassnahmen hinzugefügt	ds
23.09.09	0.3	Arbeitspakete definiert	dh
23.09.09	0.4	Risikomanagement erstellt und Massnahmen definiert	ds
23.09.09	0.5	Qualitätsmassnahmen, Review	dh
23.09.09	1.0	Review	ds
24.09.09	1.1	Review, Schreibfehler korrigiert	dh
25.09.09	1.2	Überarbeitung Arbeitspakete	ds
26.10.09	1.3	Anpassungen Konfigurationsverwaltung etc.	ds
29.11.09	1.4	Überarbeitung/Anpassung Konfigurationsverwaltung, Referenzen	dh
09.12.09	1.5	Überarbeitung, Anpassung	ds
15.12.09	1.6	Überarbeitung, Verbesserung	ds

1.2 Referenzen

Nr.	Dokument	Autor
1	SA-Häfliger-Süsstrunk-CS-FX Aufgabenstellung.pdf	ms
2	Projektantrag.pdf	Team
3	Internal Coding Guidelines http://blogs.msdn.com/brada/articles/361363.aspx	Brad Adams

2 Einführung

2.1 Zweck

Dieses Dokument dient zur Beschreibung des Projektplans für die Semesterarbeit Foreign Exchange User Interface.

2.2 Gültigkeitsbereich

Dieses Dokument bildet die Grundlage dieser Arbeit und ist deshalb über die gesamte Projektdauer des Foreign Exchange User Interface-Projektes gültig.

2.3 Definitionen und Abkürzungen

Sämtliche Definitionen/Abkürzungen, welche in der Semesterarbeit auftreten, werden in einem externen Dokument (Glossar.pdf) beschrieben. Unter anderem auch die verwendeten Finanz-Begriffe.

2.4 Übersicht

Das Kapitel 4 „Projekt Übersicht“ beschreibt Ziel und Zweck der Semesterarbeit Foreign Exchange User Interface. Nachfolgend werden Projektorganisation, Managementabläufe wie Zeitplanung, Meilenseine, Iterationen sowie das Risikomanagement beschrieben. Weiter werden die verschiedenen Arbeitspakete definiert, aus welchen sich die Arbeitsaufteilung ergibt. Im letzten Teil des Dokuments werden Infrastruktur sowie Qualitätsmassnahmen der Arbeit beschrieben.

3 Projekt Übersicht

Eine Übersicht über das Projekt befindet sich in den zwei folgenden Dokumenten:
SA-Häfliger-Süsstrunk-CS-FX Aufgabenstellung.pdf[1]
Projektantrag.pdf[2]

3.1 Zweck und Ziel

Die alte, mit Java Applet realisierte Version, welche die Credit Suisse heute verwendet, soll ein grafisch moderneres Erscheinungsbild erhalten. Die Technologie sollte auch dem heutigen „State-of-the-Art“ entsprechen.

3.2 Annahmen und Einschränkungen

Pro Woche wird pro Teammitglied rund 16 Stunden gearbeitet.

4 Projektorganisation

4.1 Team

- Daniel Häfliger dhaeflig@hsr.ch
- Dominik Süsstrunk dsuesstr@hsr.ch

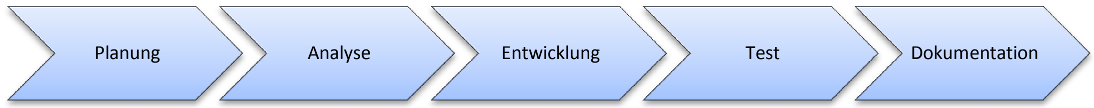
4.2 Betreuer

- Prof. Dr. Markus Stolze mstolze@hsr.ch

4.3 Industriepartner

- Dr. Beat Liver, Credit Suisse beat.liver@credit-suisse.com

4.4 Organisationsstruktur



4.4.1 Planung

- Prozessverantwortlicher
  - Dominik Süsstrunk
- Artefakte
  - Projektplan
  - Anforderungsspezifikationen

4.4.2 Analyse

- Prozessverantwortlicher
  - Dominik Süsstrunk
- Artefakte
  - Domainanalyse

4.4.3 Entwicklung

- Prozessverantwortlicher
  - Daniel Häfliger
- Artefakte
  - Code
  - SAD
  - Releases

4.4.4 Test

- Prozessverantwortlicher
  - Dominik Süsstrunk
- Artefakte
  - Code (Unittests)
  - Testprotokolle

4.4.5 Dokumentation

- Prozessverantwortlicher
  - Daniel Häfliger
- Artefakte
  - Handbuch
  - Summary

5 Management Abläufe

5.1 Projekt Kostenvoranschlag

Das Projekt hat eine Laufzeit von 14 Wochen, da für die Semesterarbeit 8 ECTS Punkte angerechnet werden und jeder ECTS Punkt ca. 2h Arbeit pro Woche ergibt. Entspricht dies ca. 230h pro Teammitglied. In der Summe also 460h.

5.2 Projektplan

5.2.1 Zeitplan

Der Zeitplan für die Arbeit befindet sich in der Datei *„02_Projektplan\Zeitplan.xlsx“*

5.2.2 Iterationsplanung

Iteration	Beschreibung	Ende
Inception	- Projektantrag - Dokumentvorlagen	SW01
Elaboration 1	- Projektplan (Zeitplan, Risiko Management, Qualitätsmerkmale) - Funktionale/nichtfunktionale Anforderungen - Use Cases Brief - Supplementary Specification - Domain Model - Systemsequenzdiagramme - External Design - Operation Contracts	SW04
Elaboration 2	- Design Model - Logische Architektur - Testbed - Architekturprototyp - Use Cases Fully Dressed - Review External Design - UI - Internal Design - Prototyp	SW06
Construction 1	- Testfälle definieren - Datenbankanbindung - Silverlight Controls erstellen - EBS Chart - Graphen - Erfassung Limitierter Aufträge - Prototyp	SW09
Construction 2	- Personalisieren der Oberfläche - Ausbau Datenbankanbindung - UI erweitern - Beta Release	SW11
Construction 3	- Abschluss - Security (Optional) - Netzwerk Security - User Levels - Release Candidate	SW13
Transition	- Benutzerdokumentation	SW14

SW: Semester Woche

5.2.3 Meilensteine

MS	Titel	Woche	Termin	Beschreibung
MS1	Projektplan	SW02	25.09.09	- Projektplan Version 1.0 - Zeitplan - Arbeitspakete definiert - Verantwortlichkeiten zugeteilt <u>Abgabe</u> - Projektplan
MS2	Analyse	SW05	16.10.09	- Dokumente des OOA - Eingearbeitet in architekturrelevante Problembereiche <u>Abgabe</u> - OOA Dokumente - Paper Prototype
MS3	Architekturprototyp	SW06	23.10.09	- Testbed läuft - Architekturprototyp - Review <u>Abgabe</u> - Architekturprototyp - Präsentation des Architektur-prototyps
MS4	Prototyp mit Custom Silverlight Controls	SW09	13.11.09	- Datenbankanbindung - Silverlight Controls - EBS Chart - Graphen - Prototyp <u>Abgabe</u> - Prototyp
MS5	Beta Release	SW11	27.11.09	- Beta Release - Personalisierbare Oberfläche - Trading - Erfassung limitierter Aufträge (OCO/OSO) <u>Abgabe</u> - Beta Release
MS6	Release Candidate	SW13	11.11.09	- Erweiterung der Erfassung limitierter Aufträge (OCO/OSO) - SAD fertig - Security (Optional) - Release Candidate <u>Abgabe</u> - SAD - Release Candidate
MS7	Abgabe	SW14	18.11.09	- Bericht - Benutzerdokumentation <u>Abgabe</u> - Benutzerdokumentation - Release Candidate - Jeglicher Dokumente - Jeglicher Code

5.2.4 Besprechungen

Alle Besprechungen werden vom Team mit einer Traktandenliste vorbereitet und die Ergebnisse in einem Protokoll dokumentiert, welches dem Betreuer und dem Industriepartner per E-Mail zugestellt wird.
Mit dem Betreuer finden in der Regel wöchentliche Besprechungen statt. Zusätzliche Besprechungen werden nach Bedarf durchgeführt. Besprechungen mit dem Industriepartner werden nach Bedarf vereinbart und durchgeführt.

5.2.5 Abgabe

Titel	Woche	Termin	Beschreibung
Architekturprototyp	SW06	23.10.09	- FOREX Quotes von Stream lesen & anzeigen - Abonnieren verschiedener FOREX Streams
Prototyp mit Custom Silverlight Controls	SW09	13.11.09	- Silverlight Controls für Anzeige der Daten - EBS Chart, Kachelansicht der abonnierten FOREX Streams, mit Statusupdate - Graph, Währungspaar Linechart der letzten Minuten, Stunden
Beta Release	SW11	27.11.09	- Oberfläche personalisierbar - Anordnung der Kacheln in EBS Chart - Einzelne Kacheln detaillierter dargestellt - Trading, Handel von Währungen - Limitierte Aufträge erfassen (OSO/OCO)
Release Candidate	SW13	11.12.09	- Erweitern des Erfassens limitierter Aufträge (OSO/OCO) - Sicherheit implementiert - Anmeldung an Server verschlüsselt - Restl. Datenübertragung verschlüsselt
Abgabe	SW14	18.12.09	- Schlussbericht

6 Risiko Management

ID	Risiko	Auswirkung	Massnahmen	Kosten	Max. Schaden	Wahrschein-lichkeit	Gewichteter Schaden	Priorität
01	Probleme mit Server/Client Architektur und Services	Verzögerung im Projektverlauf	Gute Einarbeitung in Silverlight, WCF, .NET	3h	20h	25%	5h	hoch
02	Unzureichende Dokumentation der Technologien	Längere Einarbeitungszeit	Weitere Dokumentationen/ Beispiele suchen und studieren. Bücher bestellen.	-	10h	20%	2h	mittel
03	Ausfall eines Teammitglieds infolge Krankheit/Unfall	Verzögerung im Projektverlauf	Zeitreserve einplanen	-	20h	10%	2h	mittel
04	Wissen konzentriert auf einzelne Person	Einzelne Aufgaben können nicht erfüllt werden	Wissen weitergeben in Sitzungen	4h	10h	10%	1h	niedrig
05	Probleme mit Konfigurationsverwaltung	Teile der Arbeit gehen verloren.	Einsatz von Versionierungssystem	5h	10h	10%	1h	niedrig
06	Anforderungen des Projekts ändern sich	Überarbeitung von Projektdokumenten	Regelmässige Sitzungen / Zeitreserve einplanen	-	10h	20%	2h	hoch
07	Probleme mit Foreign Exchgange Prozessen	Verzögerung im Projektverlauf	Gute Einarbeitung in Begriffe und gutes Verständnis der Geschäftsabläufe	-	20h	25%	5h	mittel
	Total Kosten in Arbeitspaketen enthalten			**12h**				
	Total Rückstellungen				**100h**		**18h**	

7 Arbeitspakete

	Pakete	Beschreibung	Soll[h]	Ist[h]
1	**Projekt Management**		**29**	**32**
1.1	Projektantrag	Festlegung, was in dem Projekt geleistet werden soll.	2	2
1.2	Projektplan	Zeitplanung, Zuweisung der Verantwortlichkeiten, Milestones festlegen und Risiken erfassen.	15	13
1.3	Zeitplan	Planung der Arbeitsstunden pro Pakete, sowie Soll/Ist Übersicht.	5	6
1.4	Konfigurationsverwaltung	MS Team Foundation Server einrichten und Build-Automatisierung.	5	9
1.5	Review		2	2
2	**Requirements**		**22**	**21**
2.1	Anforderungsspezifikation	Erfassen der Anforderungen.	8	7
2.2	Use Cases Brief	Wichtigste Use Cases im Brief Format.	2	2
2.3	Use Cases Fully Dressed	Wichtigste Use-Cases Fully Dressed.	5	4
2.4	Szenarios	Abläufe von Benutzern.	4	3
2.5	Review		3	5
3	**Analyse**		**48**	**32**
3.1	Domain Model	Domain Model anhand der Use Cases und der Spezifikationen erstellen.	15	9
3.2	Contracts	Verfassen der Contracts für die System-operationen.	6	5
3.3	SSD	Erstellen der wichtigsten Sequenz-diagramme.	4	2
3.4	Paperprototype	Handschriftliche Prototypen des User Interfaces.	15	11
3.5	Architekturanalyse	Verbindung zwischen Silverlight Applikation und Streaming Server	4	4
3.6	Review		4	0
4	**Design**		**16**	**19**
4.1	Klassendiagramm	Klassendiagramme erstellen.	8	13
4.2	Serverarchitektur	Architektur der Server/Client Verbindung beschreiben.	4	3
4.3	Review		4	2
5	**Implementation**		**185**	**173**
5.1	User Interface	Erstellen des User Interfaces	40	43
5.2	Silverlight Controls	Erstellen einzelner Controls.	30	48
5.3	Styles	Styles definieren, um ein einheitliches Erscheinungsbild zu schaffen.	5	15
5.4	Testbed	Aufsetzen eines Testbeds, welches Daten über Devisen streamt.	30	4
5.5	Dataservice	Service auf dem Testbed entwickeln, welcher die Daten bereitstellt.	20	21
5.6	Datenbank	Datenbank für die Benutzerverwaltung anlegen.	10	17

5.7	Orders	Limitierte Aufträge (OCO/OSO) erfassen, bearbeiten und löschen.	25	15
5.8	Security	Sicherheit implementieren, Verschlüsslung von Daten.	15	0
5.9	Review		10	9
6	**Test**		**30**	**40**
6.1	Usability Test	Test, ob Applikation intuitiv bedien bar ist.	15	14
6.2	Unit Test	Testmethoden im Code, um Methoden zu prüfen.	5	15
6.3	System Test	Gesamttests -> Ablauf simulieren.	10	11
7	**Dokumentation**		**59**	**98**
7.1	SAD	Architekturdokument bearbeiten.	20	33
7.2	Benutzerhandbuch & Video	Handbuch & Video(s) über die Bedienung der Applikation.	10	8
7.3	Code	Dokumentieren des Codes.	4	22
7.4	Tests	Testfälle dokumentieren – erstellen, testen, Testergebnis dokumentieren.	10	7
7.5	Summary	Kurzbeschreibung des Projektes für das WIKI von Herrn Prof. Dr. Stolze	4	7
7.7	Review		6	21
8	**Studium Technologien**		**40**	**48**
8.1	Einarbeitung Silverlight	Erlernen der Silverlight Technologie.	20	26
8.2	Einarbeitung FOREX	Verstehen, wie der Devisenhandel funktioniert.	10	10
8.3	Serverarchitektur (Streaming)	Streaming Technologien kennenlernen.	10	12
9	**Sitzungen**		**32**	**47**
9.1	Teammeetings	Häfliger/Süsstrunk	14	30
9.2	Meetings mit Betreuer	Häfliger/Süsstrunk/Stolze	14	13
9.3	Meetings mit Auftraggeber und Betreuer	Häfliger/Süsstrunk/Stolze/Liver	4	4
10	**Qualitätssicherung**		**13**	**15**
10.1	Risk Management	Auflistung möglicher Risiken und deren Folgen auf das Projekt.	3	1
10.2	Code Reviews	Quellcode durchsehen und refactorisieren.	10	14
	Total		**473**	**525**

8 Infrastruktur

8.1 Räumlichkeiten

- Gemeinsam wird an den uns zugeteilten Stationen im Zimmer 1.258 gearbeitet. Meetings werden in freien (reservierten) Sitzungsräumen an der HSR abgehalten.

8.2 Hardware

- Arbeiten an eigenem Notebook oder der zugeteilten Station
- Der Server ist ein von der Schule zur Verfügung gestellter Virtueller Server

8.3 Software

8.3.1 Test Server

- Windows Server 2003
- SVN (VisualSVN)
- Hudson
- MS SQL Server 2005

8.3.2 Programmiersprachen

- C#
- XAML (XML)

8.3.3 Entwicklungsumgebung, Tools

- Visual Studio Team System 2008
- Expression Blend 3
- Silverlight Toolkit
- Silverlight SDK
- NUnit

8.3.4 Versionsverwaltung, Automatisierung

- Versionskontrolle
    - SVN
    - VisualSVN – SVN Plugin für SVN
- Buildserver
    - Hudson mit MSBuild

8.3.5 Dokumentation

- MS Office
- Enterprise Architect

8.3.6 Visual Studio Team System

Wir wollten unsere Arbeit mit dem Visual Studio Team System verwalten, allerdings war der uns zur Verfügung gestellte Server nicht genügend leistungsstark, um ein schnelles Versionieren zu ermöglichen. Die Check-In dauerten gut zwei Minuten für einige Kilobyte an Daten. Deswegen sind wir auf SVN umgestiegen.

9 Qualitätsmassnahmen

9.1 Dokumentation

Es wird darauf geachtet, dass die Dokumentationen aktuell gehalten werden und die Änderungsgeschichte sauber angepasst wird, damit sich jeder Projektteilnehmer auf deren Inhalt verlassen kann.

9.2 Projekt- und Zeitplan aktualisieren

Der Projektplan wird wöchentlich vom Projektleiter nachgeführt und angepasst, damit er immer auf einem möglichst aktuellen Stand ist.
Jedes Mitglied trägt, nach erfolgter Arbeit, seine Zeiten in den Zeitplan ein, damit auch dieser aktuell ist und allfällige Änderungen im Projektplan klar sind. Jede Iteration wird am Ende der Vorhergehenden geplant.

9.3 Sitzungsprotokolle

Alle Meetings mit dem Projektbetreuer und/oder dem Auftraggeber werden mit einer Traktandenliste vorbereitet, welche den Teilnehmern vor dem Meeting gesendet wird. Alle Meetings werden protokolliert, damit Vorschläge, Kritik und Abmachungen schriftlich erfasst sind.

9.4 Dokumentvorlagen

Für alle Dokumentationen wird die Vorlage „*99_Templates/document_template.dotx*“ verwendet.
Für Sitzungsprotokolle wird die Vorlage „*99_Templates/metting_template.dotx*“ verwendet.

9.5 Versionsverwaltung

9.5.1 Dokumentenversionen

Die Versionierung der Dokumente sieht wie folgt aus: <Major-Version>.<Minor-Version>
Die 1. Version von Dokumenten ist immer 0.1, die Minor-Version wird bei jeder Anpassung erhöht.
Die Major-Version wird dann erhöht, wenn das Dokument zur Abgabe bereit ist.

9.5.2 Code Verwaltung

Der Code wird mit SVN verwaltet und versioniert.

9.6 Automatisierung

Auf dem Server, auf welchem der Hudson Build Server läuft, wird ein Buildjob erfasst, welcher automatisch alle 4 Stunden ausgeführt wird und die Build-Reports per E-Mail ans Team schickt.

9.7 Tests

9.7.1 System Tests

Systemtests werden, basierend auf den Use Cases, durchgeführt, protokolliert, ausgewertet und zeigen, ob die Software den funktionalen Anforderungen genügt.

9.7.2 Usability Tests

Externe Benutzer (potentielle Benutzer des Systems) testen die Prototypen auf ihre Benutzerfreundlichkeit und geben Feedback, welches dann ausgewertet wird.

9.8 Codierungs Richtlinien

Um das gemeinsame Arbeiten zu erleichtern, halten sich alle Teammitglieder an folgende Punkte betreffend dem Umgang mit Codefiles. Grundsätzlich basieren diese Vorgaben auf dem Dokument [Internal Coding Guidelines] von Brad Adams[3].

9.8.1 Kommentare

Alle public Methoden müssen kommentiert werden. Bei Methoden muss nur dokumentiert werden, **was** die jeweilige Methode macht und nicht **wie**.

9.8.2 Namensgebung

- Klassen und Methoden haben einen Namen, welche ihre Aufgabe widerspiegelt.
- Klassen und Methoden müssen mit einem Grossbuchstaben beginnen. Interfaces mit einem I
- Der Name der Assemblies widerspiegelt die implementierten Klassen und deren Namespace.
- Namen sollen in Camel-Case geschrieben werden.

9.8.3 Formatierung

- Geschweifte Klammern sollen auf einer separaten Zeile geschrieben werden.
- Im Allgemeinen sollten die vorgegebenen Einstellungen im Visual Studio beibehalten werden.

9.9 Reviews

9.9.1 Dokumentreviews

Vor einer Abgabe werden die Dokumente von beiden Teammitgliedern durchgelesen und die Fehler korrigiert. Danach wird die Major-Version des Dokuments erhöht, sowie die Minor-Version zurückgesetzt.

9.9.2 Codereviews

Bei den Codereviews wird vor allem darauf geachtet, dass der Quellcode den Richtlinien entspricht und mögliche Codesmells verschwinden. Demnach ist der Ablauf eines Codereviews: Codesmells finden und diese refaktorisieren.

Die Reviews des Codes werden jeweils von dem Teammitglied, welches weniger an der Quellcode-Datei gearbeitet hat, vorgenommen.

Foreign Exchange UI

Anforderungsspezifikationen

1 Dokumentinformationen

1.1 Änderungsgeschichte

Datum	*Version*	*Änderung*	*Autor*
23.09.09	0.1	Dokumentstruktur erstellt	ds
25.09.09	0.2	Grundlegendes erfasst	ds
29.09.09	0.3	Lizenzanforderungen	ds
30.09.09	0.4	Use Case Diagramm / Use-Cases Brief	dh
02.10.09	0.5	Schreibfehler verbessert/Use-Case Fully Dressed	ds/dh
06.10.09	1.0	Review	ds
07.11.09	1.1	Überarbeitung	dh
09.12.09	1.2	Überarbeitung	ds
15.12.09	1.3	Überarbeitung / Verbesserung	ds

1.2 Referenzen

Nr.	*Dokument*	*Autor*
1	Java Applet Version der bestehenden Anwendung https://entry.credit-suisse.ch/csfs/p/cb/de/online/onl_uebersicht.jsp	CS
2	Silverlight 3.0 http://www.silverlight.net	Microsoft

2 Allgemeine Beschreibung

2.1 Produkt Perspektive

Die CS hat eine bestehende Foreign Exchange Online Software[1] auf Basis von Java Applets. Da neue Technologien wie Silverlight, Flex oder GWT im Anmarsch sind, will man da die Möglichkeiten austesten. Wichtig ist hier vor allem das Anzeigen des EBS-Charts und Erfassen von Aufträgen. Weiter soll das Produkt personalisierbar sein und auf die Bedürfnisse des Kunden angepasst werden können.
Die Sicherheit soll anfangs vernachlässigt werden.

2.2 Produkt Funktion

- Anzeigen des EBS Charts
    - EBS Ansicht für Currency Pair
    - Eigene Ansichten erstellen
    - LineChart für Currency Pair mit History anzeigen
- Erfassen von einfachen Aufträgen
    - Spots
    - Outright
    - Swap
- Erfassen von limitierten Aufträgen
    - Erfassen von Call-Levels in LineChart

2.3 Benutzer Charakteristik

Es gibt tendenziell zwei verschiedene Benutzergruppen. Den Amateur- und den Profitrader.

2.3.1 Amateur

Der Amateur braucht das Programm nicht allzu oft. Da er nicht täglich mit der Software arbeitet und auch nicht täglich „tradet“, ist er eher unsicher. In der bestehenden Lösung sind für den Amateur sogenannte Zeitscheiben untergebracht. Wenn er einen Trade machen will, erscheint eine Sicherheitsabfrage mit einer Zeitvorgabe, wie lange er für die Entscheidung noch Zeit hat.

2.3.2 Profi

Der Profi arbeitet eigentlich immer mit der Software und bewegt sich auch sonst viel in der FOREX-Welt. Er ist sich seiner Entscheidung sicher. In der bestehenden Lösung sind hier sogenannte Sicherheitsklicks eingebaut. Will der Trader eine Order auslösen, so tut er dies zum Beispiel mit ctrl + click.

2.4 Einschränkungen

- Um die Software ausführen zu können, muss die neuste Version von Silverlight[2] auf dem Clientbrowser installiert sein.
- Der Client muss eine Verbindung zum Server haben (Inter-/Intranet).

2.5 Annahmen

- Die Applikation wird immer werktags ausgeführt.
- Der Benutzer besitzt ein Login.

2.6 Abhängigkeiten

- Allgemeine Foreign Exchange Aufträge müssen den originalen Auftragsarten entsprechen
- Da die Software mit Silverlight umgesetzt wird, ist sie stark an das .NET Framework gebunden

3 Spezifische Anforderungen

3.1 Funktionale Anforderungen

3.1.1 Angemessenheit

Da dies eine Internetanwendung ist und der Benutzer beim Starten die Software immer komplett aus dem Internet herunterladen muss, sollte diese mehr als 25 MB gross sein, damit es nicht zu langer Wartezeit kommt.

3.1.2 Richtigkeit

Da es beim Foreign Exchange um grössere Geldbeträge geht, müssen die Daten zwingend richtig weiterverarbeitet werden vom System.

3.1.3 Sicherheit

Anfangs soll nur ein einfaches Benutzerlogin zur Sicherheit beitragen. Später können immer noch weitere Sicherheitsmechanismen eingeführt werden, wie z.B. die Verschlüsslung der Daten über das Netz etc.

3.2 Bedienbarkeit

3.2.1 Wiederherstellbarkeit

Die Daten müssen auch bei einem Browserabsturz oder versehentlichem Beenden wieder verfügbar sein. Es dürfen keine Aufträge verloren gehen.

3.2.2 Erlernbarkeit

Die Applikation soll schnell und einfach zu erlernen sein. Benutzer sollen nötige Instruktionen direkt innerhalb der Applikation z.B. über Video erhalten.

3.2.3 Verständlichkeit

Das GUI soll intuitiv zu bedienen sein. Benutzer, die schon die alte Version[1] benutzt haben, sollen sich möglichst rasch und einfach auf dem neuen GUI zurechtfinden können.

3.3 Zuverlässigkeit

3.3.1 Verfügbarkeit

Die Software muss zu 99.9% verfügbar sein, da es sich hier, wie bereits erwähnt, um grosse Geldbeträge handelt.

3.3.2 Fehlertoleranz

Bei schlechter Netzverbindung dürfen keine Daten verloren gehen.

3.4 Leistung

3.4.1 Latenz- / Antwortzeiten

Möglichst geringe Latenz- sowie Antwortzeiten, da sich die Kurse bei jeder Transaktion ändern.

3.5 Lizenzanforderungen

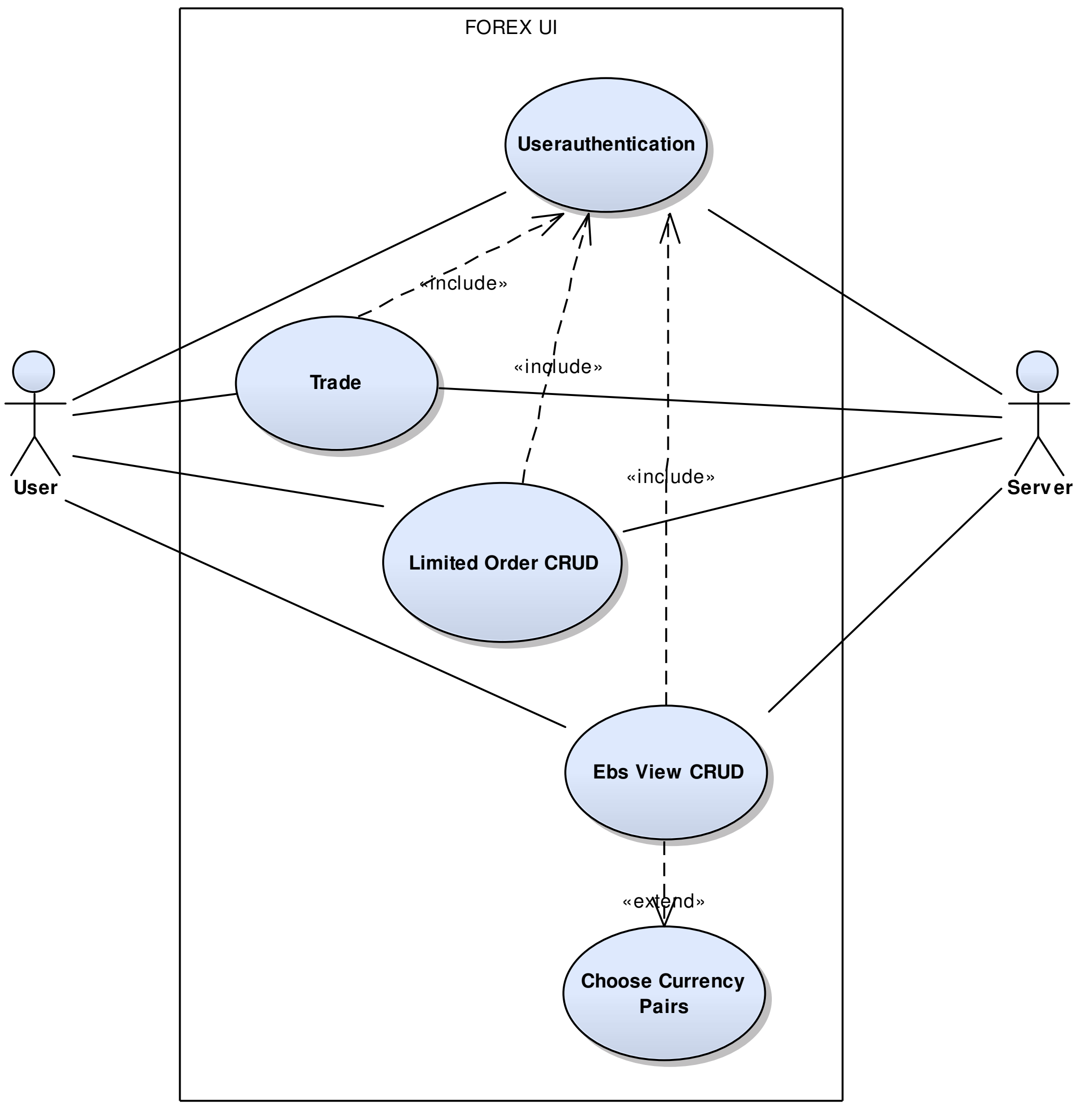
Die Rechte an der Software gehören

1. den Entwicklern Daniel Häfliger und Dominik Süsstrunk
2. der HSR
3. der Credit Suisse

Die Software kann von allen Rechteinhabern so weiterverwendet werden, wie diese es wollen.

4 Use Cases

4.1 Use Case Diagramm



4.2 Aktoren & Stakeholders

Die Aktoren sind Benutzer. Diese beinhalten Bankangestellte sowie Bankkunden, welche die FOREX Applikation benutzen.

4.3 Userauthentication

Overview	Benutzer authentifizieren / am FOREX UI anmelden
Stakeholder and Interests	Benutzer: Möchte die Anwendung nutzen
Preconditions	Silverlight Plug-In für verwendeten Webbrowser installiert
Success Guarantee	Der Benutzer ist angemeldet und kann das FOREX UI nutzen
Main Success Scenario	1. Der Benutzer öffnet die FOREX Trading Website mit der Silverlight Applikation 2. Er meldet sich mit Benutzernamen und Passwort am System an 3. Die eingegebenen Daten werden auf dem Server authentifiziert 4. Der Benutzer ist am FOREX UI angemeldet
Extensions/Alternative Flows	a. Die Authentifizierung ist fehlgeschlagen 5. Der Benutzer wird nicht gefunden 6. Das eingegebene Passwort stimmt nicht mit demjenigen des Benutzers überein
Special Requirements	
Technology and Data Variations List	
Frequency of Occurence	Bei jedem Anwendungsstart Trader: täglich Amateur: wöchentlich
Miscellaneous	

4.4 EBS View CRUD

Overview	EBS Ansicht mit einzelnen Währungspaar-Kacheln anzeigen und die angezeigte Ansicht verwalten
Stakeholder and Interests	Benutzer: Möchte die Ansicht von Währungspaaren anpassen
Preconditions	Use Case Userauthentication abgeschlossen
Success Guarantee	Benutzer sieht EBS Ansicht so, wie er diese gerne sehen möchte und sie beim letzten Mal eingestellt hat
Main Success Scenario	1. Der Benutzer sieht die zuletzt eingestellte EBS Ansicht 2. Er bearbeitet die Ansicht, indem er die Kacheln an die gewünschte Position verschiebt
Extensions/Alternative Flows	1. a) Bei der Erstanmeldung wird dem Benutzer eine Standardauswahl angezeigt 1. b) Möchte er eine andere, gespeicherte Ansicht, ansehen, so wählt er diese in einer Liste aus 2. a) Der Benutzer fügt eine neue Kachel durch Klicken und Auswählen des Währungspaars (Sub Use Case: Choose Currency Pairs) zur Auswahl hinzu 2. b) Der Benutzer entfernt eine Kachel, indem er diese schliesst 2. c) Der Benutzer möchte eine weitere Ansicht erstellen. Er erstellt eine neue Ansicht durch Auswählen von ‚Ansicht hinzufügen‘ 2. d) Eine alte, nichtbenutzte Liste, soll entfernt werden. Der Benutzer wählt ‚Ansicht entfernen‘
Special Requirements	
Technology and Data Variations List	
Frequency of Occurence	Laden der Daten im Minimum bei jedem Anwendungsstart Trader: mehrmals täglich

	Amateur: wöchentlich
Miscellaneous	Die aktuelle Ansicht wird beim Wählen einer anderen/neuen Ansicht, sowie beim Verlassen der Applikation gespeichert

4.5 Sub Use Case: Choose Currency Pairs

Overview	Währungspaar Kacheln für die EBS Ansicht auswählen
Stakeholder and Interests	Benutzer: Möchte ein neues Währungspaar auswählen
Preconditions	EBS View CRUD – Use Case, Alternative Flow 2.a) tritt ein
Success Guarantee	Ausgewähltes Währungspaar wird verwendet
Main Success Scenario	1. Auswahl der 1. Währung durch Benutzer 2. Auswahl der 2. Währung durch Benutzer 3. Bestätigen der Eingaben
Extensions/Alternative Flows	
Special Requirements	
Technology and Data Variations List	
Frequency of Occurence	Beim Hinzufügen von neuen Währungspaaren Trader: selten (beim 1. Start, danach nur noch spärlich) Amateur: selten (beim 1. Start, danach ab und an)
Miscellaneous	

4.6 Trade

Overview	Einen Trade zwischen einem Währungspaar abwickeln
Stakeholder and Interests	Benutzer: Möchte Fremdwährungen handeln System: Möchte Auftrag erfolgreich erfassen und Provision kassieren
Preconditions	Use Case Userauthentication abgeschlossen
Success Guarantee	Eine Transaktion zum Handeln zweier Währungen wurde auf dem Server erstellt
Main Success Scenario	1. Auswahl des Währungspaares (durch Benutzer), mit welchem gehandelt werden soll 2. Eingabe des Betrages, der gehandelt werden soll 3. Durch Auswählen von ‚kaufen' für den eingegebenen Betrag Fremdwährung kaufen 4. Fenster zur Überprüfung der Transaktion, welches vom Benutzer bestätigt wird
Extensions/Alternative Flows	3. a) Durch Auswählen von ‚verkaufen' Fremdwährung des eingegebenen Betrages verkaufen 4. a) Das Angebot ist nur eine gewisse Zeit gültig. Ist diese Zeit abgelaufen, muss ein neuer Kurs geladen werden, welcher dann wieder für diese Zeitspanne gültig ist 4. b) Handelt es sich bei dem Benutzer um einen Trader, so wird dieses Fenster nicht angezeigt und der Handel sofort vollzogen
Special Requirements	
Technology and Data Variations List	
Frequency of Occurence	Bei jedem Handel von Währungen

	Trader: sehr oft (Stündlich mehrmals) Amateur: wöchentlich
Miscellaneous	

4.7 Limited Order CRUD

Overview	Erfassen von limitierten Aufträgen und Verwalten von aktiven limitierten Aufträgen
Stakeholder and Interests	Benutzer: Möchte limitierte Aufträge verwalten
Preconditions	Use Case Userauthentication abgeschlossen
Success Guarantee	Limited Order ist erstellt, angepasst oder abgebrochen worden.
Main Success Scenario	1. Auswahl des Währungspaares (durch Benutzer), mit welchem gehandelt werden soll 2. Eingabe des Limited Order Typs 3. Eingabe des Betrages, der gehandelt werden soll 4. Limits und deren Art (SL/TP) erfassen 5. Ablaufdatum/Zeit erfassen 6. Zone auswählen 7. Limited Order aktivieren 8. Fenster zur Überprüfung der Transaktion, welches vom Benutzer bestätigt wird
Extensions/Alternative Flows	2. a) Bestehenden Limited Order zum Bearbeiten öffnen 2. b) Bestehenden Limited Order abbrechen 3. a) Verschiedene Beträge eingeben
Special Requirements	
Technology and Data Variations List	
Frequency of Occurence	Trader: sehr oft (Täglich mehrere Male) Amateur: selten (Monatlich)
Miscellaneous	

Foreign Exchange UI

Szenarios

1 Dokumentinformationen

1.1 Änderungsgeschichte

Datum	Version	Änderung	Autor
25.09.09	0.1	Dokument erstellt / Personas	dh
30.09.09	0.2	Szenarios verfasst	dh
06.10.09	1.0	Review	ds
15.12.09	1.1	Überarbeitung/Verbesserung	ds

2 Personas

2.1 Oskar

Oskar ist ein 43 jähriger Banker, der im Devisenhandel arbeitet. Er benutzt während der Arbeit stets einen Computer mit der FOREX-Software der Bank. Das Zehnfinger-System hat er nie richtig gelernt, aber er benötigt es bei seiner Arbeit auch nicht zwingend. Oskar arbeitet gerne; am Morgen zählt er zu den Ersten und am Abend bleibt er manchmal länger. Zuhause besitzt er einen Computer, benutzt ihn aber nur selten, um seine privaten E-Mails abzurufen. Sein Sohn Lukas (12) zeigt ihm jedoch immer wieder, was er im Internet so alles entdeckt hat.

2.2 Patrik

Patrik ist ein 29 jähriger, aufstrebender Banker. Seit dem Abschluss seines Wirtschaftsstudiums ist er als Kundenberater im Investmentbereich einer Bank tätig. Seine Arbeit erledigt er stets gewissenhaft und sauber. Er beherrscht den Computer, auch wenn er ihn während der Arbeit nicht so oft gebraucht. Zuhause interessiert er sich für Wertpapiere wie auch für den Devisenhandel. Auch wenn sein Kapital noch nicht so gross ist, handelt er immer wieder mit kleineren Beträgen über das Online-Portal der Bank.

2.3 Otto

Otto, 52 jährig, Inhaber eines Elektronikgeschäftes, welches Platinen für Grossprojekte herstellt und bestückt. Die Elektronikkomponenten kauft er über einen langjährigen Handelspartner in den USA ein, auf welchen er sich verlassen kann. Um seinen Kunden Offerten zu erstellen, welche für ihn möglichst rentabel sind, sollte er wissen, wie viel ihn die Komponenten kosten. Er möchte deshalb in Zukunft CHF in USD anlegen, so können ihm grössere Kursschwankungen nichts anhaben. Vom Devisenhandel versteht er nicht sehr viel. Er hat die, von der Bank zur Verfügung gestellten Dokumente, sowie ein Buch gelesen und möchte möglichst einfach Geschäfte abwickeln können. Vom Computer versteht er nicht allzu viel. Er beherrscht allerdings das Zehnfinger-System, um seine Offerten zu schreiben.

3 Szenarios

3.1 Oskars Arbeitstag

Morgens pünktlich um 7:30 Uhr ist Oskar an seinem Arbeitsplatz und richtet sich für seinen Tag ein. Er hat das Gefühl, dass heute der Schweizer Franken im Verlaufe des Tages ein wenig einbricht. Um die Devisen handeln zu können, meldet er sich an dem FOREX System seines Arbeitgebers an und wählt eine Ansicht auf verschiedene Devisenkurse (z.B. CHFEUR, USDCHF, GBPCHF). Bei seinem ersten Handel an diesem Tag, möchte er für 500‘000 CHF Euros kaufen. Er gibt diesen Betrag in die Applikation ein und verfolgt den Markt ein wenig. Als der Wechselkurs niedrig ist, handelt er sofort und kauft mit einem Klick auf den Kaufen-Knopf Euros im Wert von 500‘000 CHF (Kurs 1.51:1). Er hat 330'000 Euros für die CHF erhalten. Später an diesem Tag trifft Oskars Vorhersage ein; der Schweizer Franken sinkt und zu seinem Glück steigt der Wert des Euros. Er verkauft die 330‘000 Euros zu einem Kurs von 1.55:1 und erhält 511‘000 CHF – ein Gewinn von 11‘000 Schweizer Franken.

3.2 Patriks Feierabendbeschäftigung

Es ist wieder einmal 18:00 Uhr, ein langer Arbeitstag geht für Patrik zu Ende und er tritt seinen Heimweg an. Nachdem er zuhause angekommen ist und sich etwas gekocht hat, setzt er sich vor seinen Computer. Er hat im Verlaufe des Tages die Devisenkurse beobachtet und bemerkt, dass das Britische Pfund zu einem sehr tiefen Wechselkurs gehandelt wird und er vermutet, dass dieser in den nächsten Tagen markant ansteigt. Er meldet sich über das Online-Banking an der FOREX-Trading Software an und sieht die Wechselkurse. 1.55:1, das hört sich sehr verlockend an. Er gibt seinen gewünschten Betrag (20‘000 CHF) ein. Das Geld hatte er eigentlich für Ferien mit seiner Freundin gespart, allerdings ist dies nun doch zu verlockend. Er klickt den ‚buy‘-Knopf und ein Fenster öffnet sich. Er hat noch 5 Sekunden Zeit sich definitiv zu entscheiden. ‚Jetzt oder nie‘ denkt er sich und bestätigt das Fenster.
Zwei Monate später hat sich das Pfund sehr gut erholt. Der derzeitige Wechselkurs liegt bei 1.63:1, ein guter Zeitpunkt, seine 12‘900£ wieder zu verkaufen. Er meldet sich erneut bei der FOREX Applikation an und erfasst den Auftrag 12‘900 GBP für 21‘000 CHF zu verkaufen. Er hat nun 1‘000 CHF Gewinn gemacht. Nicht sehr viel, aber es reicht, um ein paar Tage länger in den Ferien zu bleiben.

3.3 Ottos Offerte

Die Firma Embedded Solutions möchte ein neues Produkt auf den Markt bringen. Um den Projektablauf sowie die Kosten möglichst gut planen zu können, holen sie eine Offerte bei Ottos Firma ein.
Die Offerte bezieht sich auf die Bestellung von 10‘000 bestückten Platinen, welche das Herz des neuen Produktes sein wird. Auf dieser Platine befinden sich ein etwas teurerer Mikrokontroller sowie etliche kleine und günstige Komponenten. Der Projektablauf sieht vor, dass die Platinen voraussichtlich in einem Jahr geliefert werden müssen.
Otto schreibt nun eine Offerte. Die 10‘000 leeren Platinen besitzt er bereits und muss diese nicht bestellen. Anders den etwas teureren Mikrokontroller sowie die zusätzlichen Komponente. Pro Platine würde ihn dies zurzeit $124.60 oder umgerechnet 131.0707 CHF kosten (Wechselkurs: 1:1.0272). Dies ergibt ein Gesamtvolumen von 1.31 Mio. CHF.

Die Prognosen für den Wechselkurs USDCHF sehen so aus, als ob dieser in einem halben Jahr bei 1:1.22 steht. Da Otto die Komponenten erst dann bestellen wird, würde ihn dies 1.52 Mio. CHF kosten. Er hätte also eine Differenz von ca. 210'000 CHF.
Um dies vorzubeugen entschliesst er sich beim Zuschlag seiner Offerte die $1.24 Mio. bereits zu wechseln, um mit diesen dann in einem halben Jahr verlustfrei die Komponenten einzukaufen.

3.3.1 Zuschlag

Otto erhält den Zuschlag der Embedded Solutions und beschliesst nun, das Auftragsvolumen für seinen Amerikanischen Handelspartner zu tauschen. Er öffnet die Devisenhandels-Plattform (FOREX UI) seiner Bank und meldet sich mit seinen Zugangsdaten an.
Er möchte nun einen Forward-Auftrag erfassen und sieht, dass er Glück hat. Der Wechselkurs, zu welchem er heute wechseln kann, steht bei 1:1.0172, was ihm über 15'000 CHF einspart und er somit noch mehr Gewinn erwirtschaften kann.
Schnell gibt er die $1.246 Mio. ein und klickt auf ‚buy'. Das Popup-Fenster, welches sich öffnet enthält alle Informationen zur Transaktion. Er bestätigt diese und die Transaktion ist erfasst.

Foreign Exchange UI

Domainanalyse

1 Dokumentinformationen

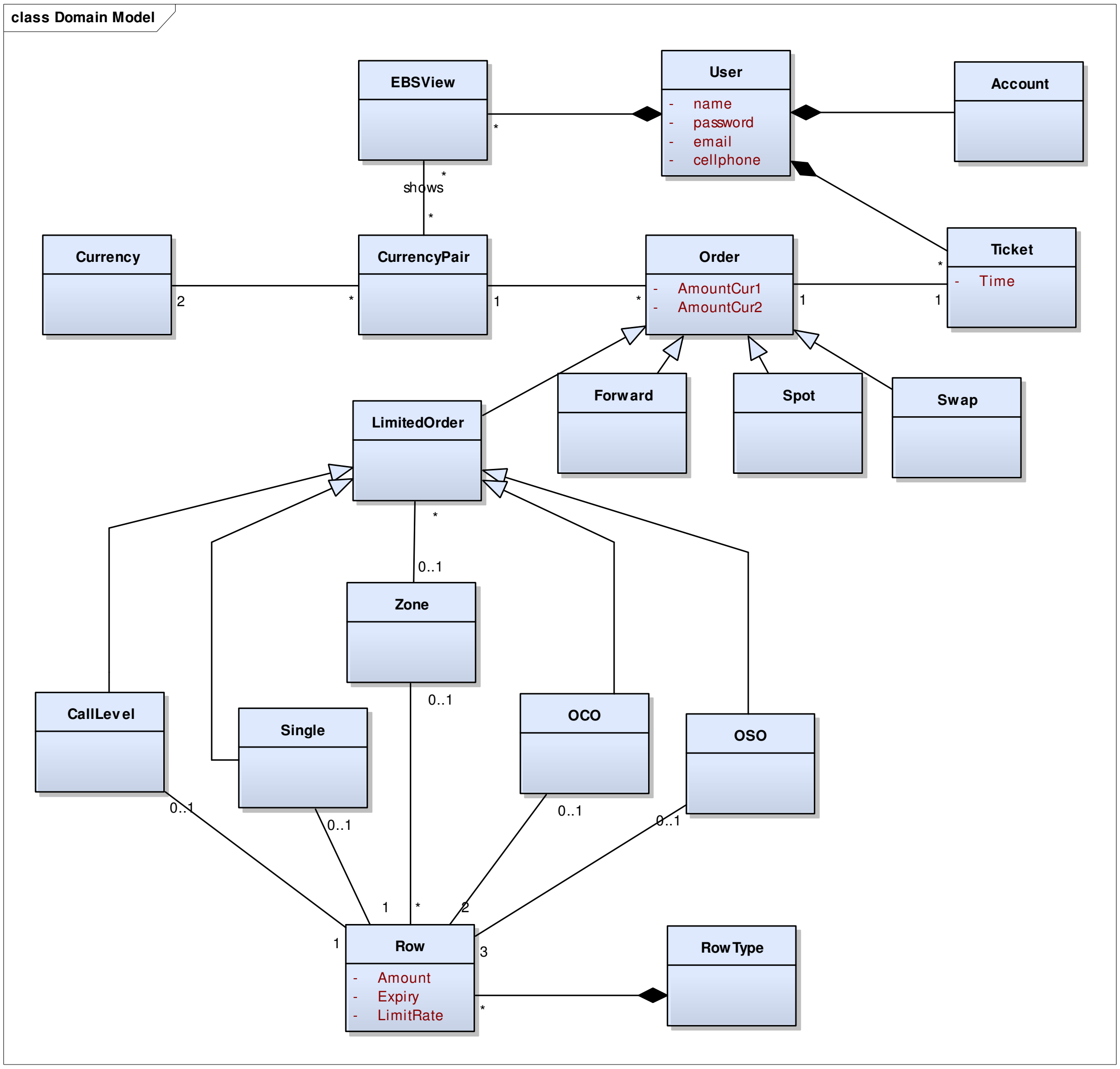
1.1 Änderungsgeschichte

Datum	Version	Änderung	Autor
02.10.09	0.1	Dokument erstellt	dh
02.10.09	0.2	Domainmodel / SSD / Contracts	dh
06.10.09	1.0	Review	ds
15.12.09	1.1	Überarbeitung / Verbesserung	ds

1.2 Referenzen

Nr.	Dokument	Autor
1	FX Expert Instruction Manual	CS

2 Domain Model



2.1 Ticket

Ein Ticket gehört zu einem Benutzer. Es wird angezeigt, wann das Ticket eröffnet wurde.
Das Ticket dient momentan nur als Schnittstelle zu den Orders.

2.2 Order

Ein Order gehört jeweils zu einem Ticket und besitzt ein Currency-Pair sowie die Beträge beider Währungen. Ein Order ist ein bestimmter Typ. Die Typen haben in sich andere Funktionalität.

2.2.1 Swap/Spot/Forward

Sind simple Aufträge, funktionsweise nachzulesen[1].

2.2.2 Limited Order

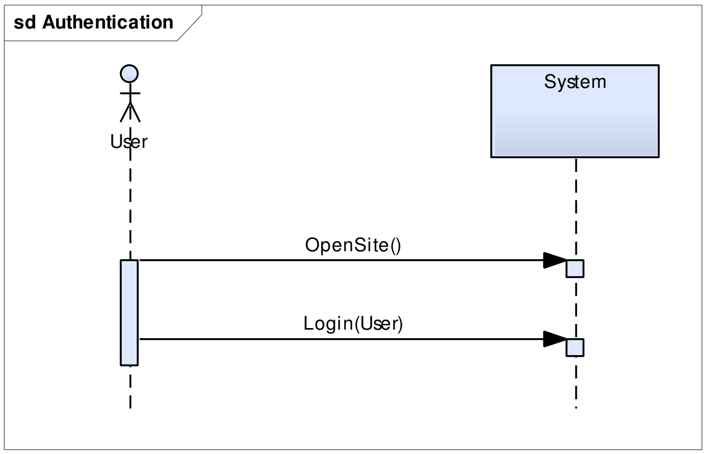
Ein limitierter Auftrag besitzt wiederum verschiedene Funktionalitäten. Je nach Funktionalität gibt es eine oder mehrere Zeilen. In einer Zeile werden Betrag, Enddatum sowie die limitierte Rate angegeben. Zudem gehört zu jeder Row, ein Zeilentyp (Take Profit/Stop Loss). Funktionsweise der einzelnen LimitedOrder-Typen nachzulesen[1].

2.3 EBSView

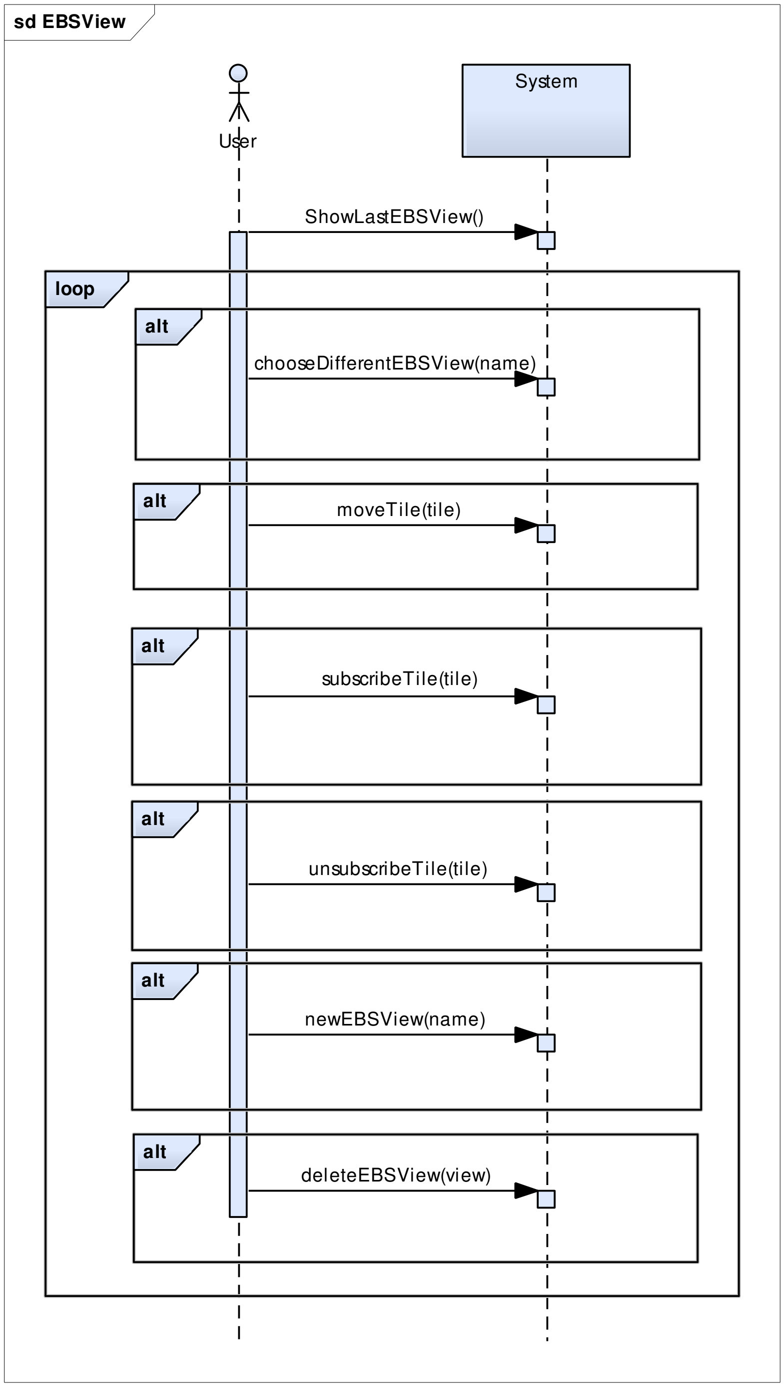
Eine EBSView ist eine Ansicht eines Benutzers auf verschiedene Währungspaare. Ein Benutzer kann mehrere solcher EBSViews besitzen.

3 System Sequenzdiagramme

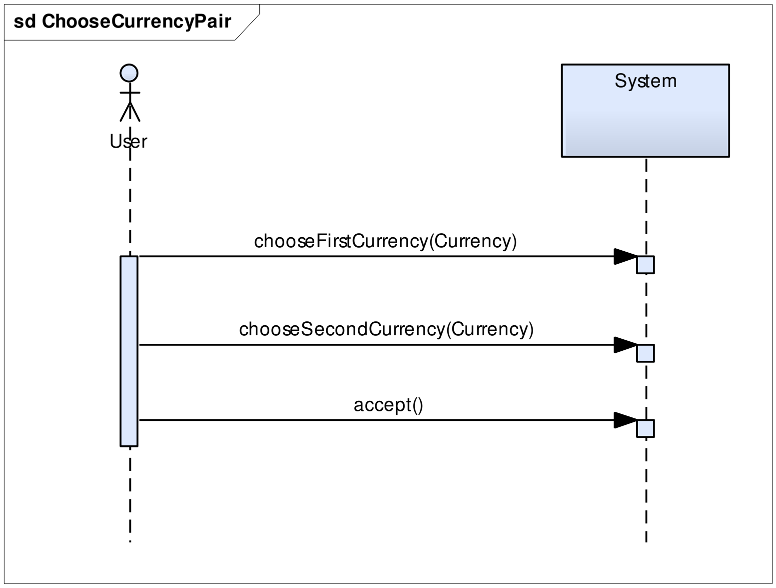
3.1 Userauthentication



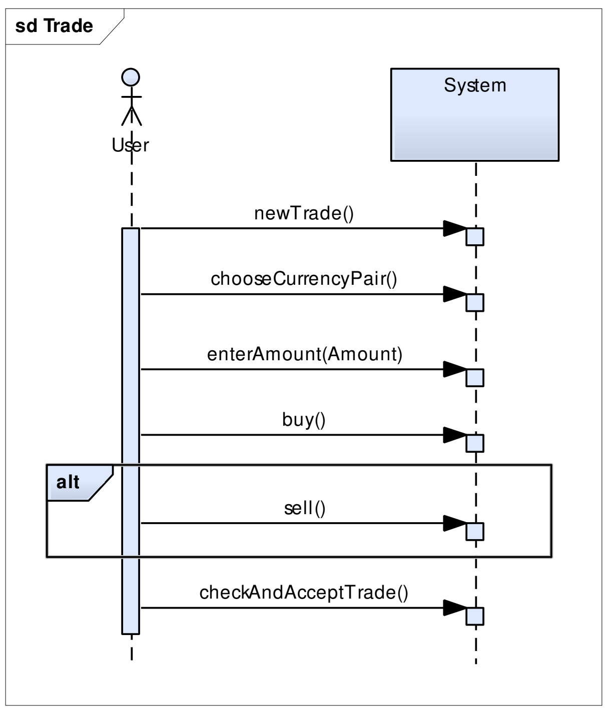
3.2 EBS View CRUD



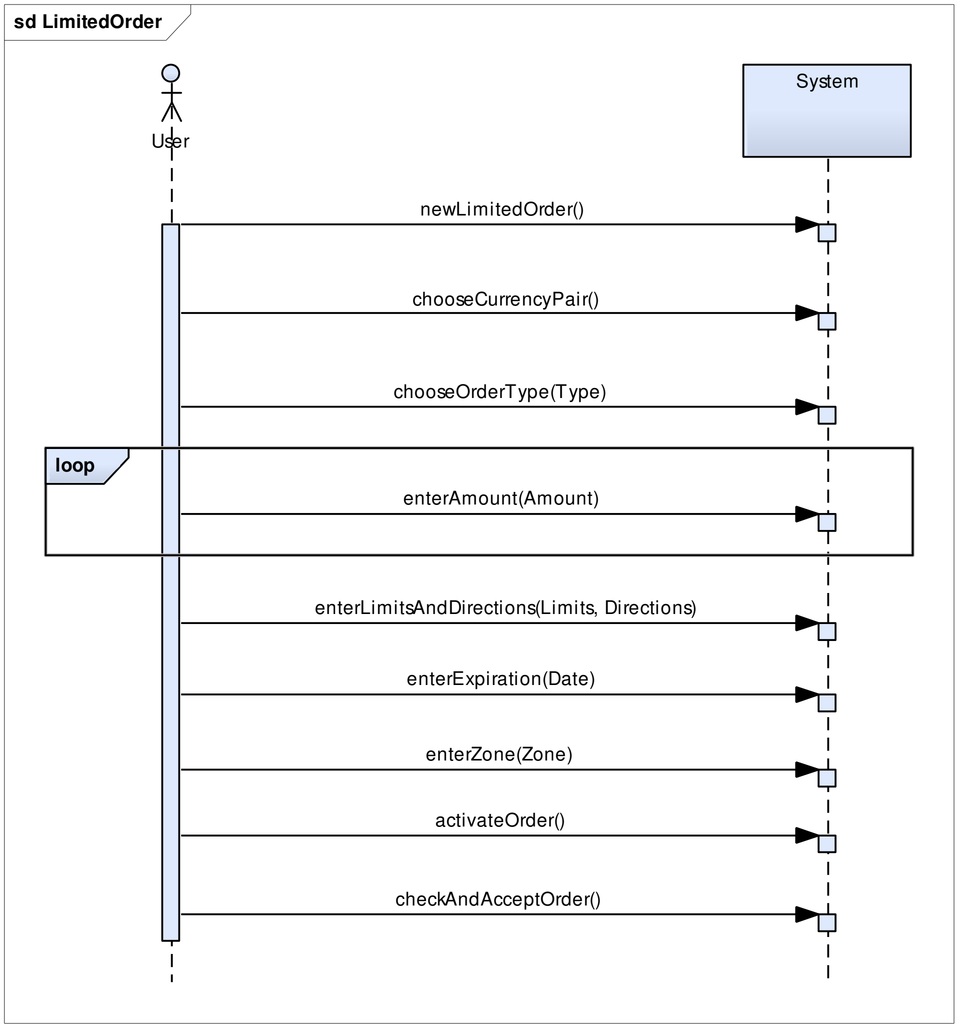
3.2.1 Sub Use Case: Choose Currency Pair



3.3 Trade



3.4 Limited Order



4 Systemoperationen

4.1 Userauthentication

4.1.1 Login

Operation	Login(user)
Vorbedingung	Silverlight Plugin Installiert Seite verfügbar
Nachbedingung	Bei richtigen Benutzerangaben angemeldet

4.2 EBS View CRUD

Vorbedingung	Userauthentication erfolgreich

4.2.1 Show Last EBS View

Operation	ShowLastEBSView()
Vorbedingung	
Nachbedingung	Anzeige des EBS-Screens, so wie ihn der Benutzer zuletzt gesehen hat oder bei der Erstanmeldung Default-Screen

4.2.2 Choose Different EBS View

Operation	ChooseDifferentEBSView(name)
Vorbedingung	
Nachbedingung	Anzeige eines anderen, gespeicherten EBS-Views

4.2.3 Move Tile

Operation	MoveTile(tile)
Vorbedingung	Mindestens zwei Tiles in EBS-View
Nachbedingung	Verschobenes Tile an neuer Position gespeichert

4.2.4 Subscribe Tile

Operation	SubscribeTile(tile)
Vorbedingung	Mindestens ein noch nicht abonniertes Tile vorhanden
Nachbedingung	Tile wird in EBS View angezeigt und gespeichert

4.2.5 Unsubscribe Tile

Operation	UnsubscribeTile(tile)
Vorbedingung	Tile bereits registriert
Nachbedingung	Tile ist von EBS View entfernt

4.2.6 New EBS View

Operation	NewEBSView(name)
Vorbedingung	EBS View mit selbem Name existiert noch nicht
Nachbedingung	Neue EBS View ist mit dem Namen angelegt und wird angezeigt

4.2.7 Delete EBS View

Operation	DeleteEBSView(name)
Vorbedingung	EBS View mit dem Namen existiert
Nachbedingung	EBS View ist entfernt und andere View wird angezeigt Wenn es die letzte EBS View war, wird eine neue, leere EBS View angelegt

	(NewEBSView)

4.2.8 Choose Currency Pair

4.2.8.1 Choose First Currency

Operation	ChooseFirstCurrency(Currency)
Vorbedingung	
Nachbedingung	1. Währung ausgewählt

4.2.8.2 Choose Second Currency

Operation	ChooseSecondCurrency(Currency)
Vorbedingung	1. Währung ausgewählt
Nachbedingung	2. Währung ausgewählt

4.2.8.3 Accept

Operation	Accept(Currency)
Vorbedingung	1. und 2. Währung ausgewählt
Nachbedingung	Währungspaar akzeptiert

4.3 Trade

Vorbedingung	Userauthentication erfolgreich

4.3.1 New Trade

Operation	NewTrade()
Vorbedingung	
Nachbedingung	Neuer Handel eröffnet

4.3.2 Choose Currency Pair

Operation	ChooseCurrencyPair()
Querverweis	4.2.8 Choose Currency Pair

4.3.3 Enter Amount

Operation	EnterAmount(Amount)
Vorbedingung	Währungspaar ausgewählt
Nachbedingung	Betrag in Trade festgelegt

4.3.4 Buy

Operation	Buy()
Vorbedingung	Amount eingegeben
Nachbedingung	

4.3.5 Sell

Operation	Sell()
Vorbedingung	Amount eingegeben
Nachbedingung	

4.3.6 Check And Accept Trade

Operation	CheckAndAcceptTrade()
Vorbedingung	Sell/Buy ausgeführt
Nachbedingung	Neue Transaktion geöffnet, Währung wird gehandelt

4.4 Limited Order

Vorbedingung	Userauthentication erfolgreich

4.4.1 New Limited Order

Operation	NewLimitedOrder()
Vorbedingung	
Nachbedingung	Neuer limitierter Auftrag eröffnet

4.4.2 Choose Currency Pair

Operation	ChooseCurrencyPair()
Querverweis	4.2.8 Choose Currency Pair

4.4.3 Choose Order Type

Operation	ChooseOrderType(Type)
Vorbedingung	Währungspaar ausgewählt
Nachbedingung	Typ des limitierten Auftrages gesetzt

4.4.4 Enter Amount

Operation	EnterAmount(Amount)
Vorbedingung	Typ des limitierten Auftrages gesetzt
Nachbedingung	Betrag in Limited Order festgelegt

4.4.5 Enter Limits and Directions

Operation	EnterLimitsAndDirections(Limits,Directions)
Vorbedingung	Betrag in Limited Order festgelegt
Nachbedingung	Limits sowie deren Directions wurden erfasst

4.4.6 Enter Expiration

Operation	EnterExpiration(Date)
Vorbedingung	Limits sowie Directions wurden festgelegt
Nachbedingung	Ablaufdatum erfasst

4.4.7 Enter Zone

Operation	EnterZone(Zone)
Vorbedingung	Ablaufdatum erfasst
Nachbedingung	

4.4.8 Activate Order

Operation	ActivateOrder()
Vorbedingung	Ablaufdatum erfasst
Nachbedingung	

4.4.9 Check And Accept Order

Operation	CheckAndAccept Order ()
Vorbedingung	Activate ausgeführt
Nachbedingung	Neue Transaktion geöffnet, Limited Order erfasst

Foreign Exchange UI

Paperprototype

1 Dokumentinformationen

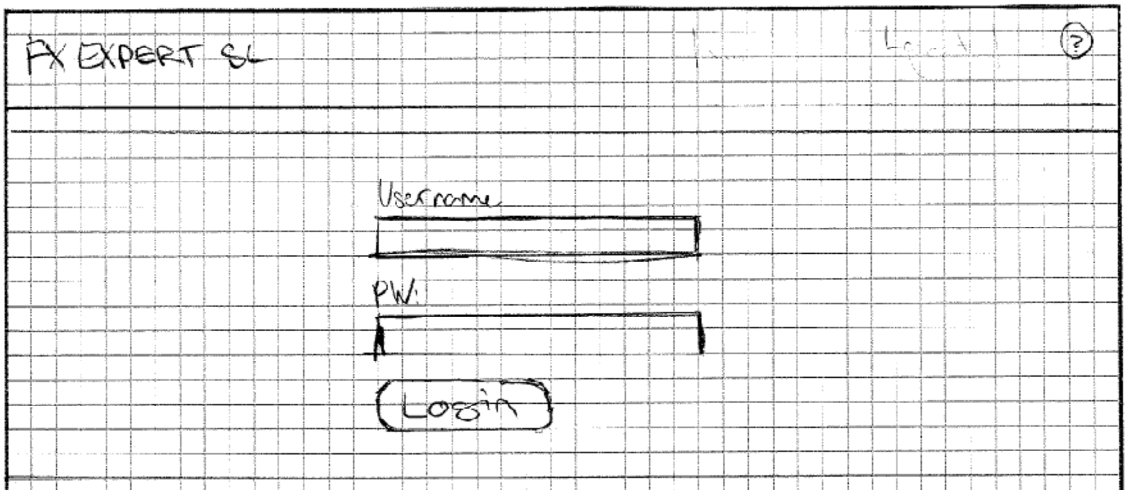
1.1 Änderungsgeschichte

Datum	*Version*	*Änderung*	*Autor*
08.10.09	0.1	Dokument erstellt; Screens hinzugefügt	ds
08.10.09	1.0	Review	Team
15.12.09	1.1	Überarbeitung / Verbesserung	ds

1.2 Referenzen

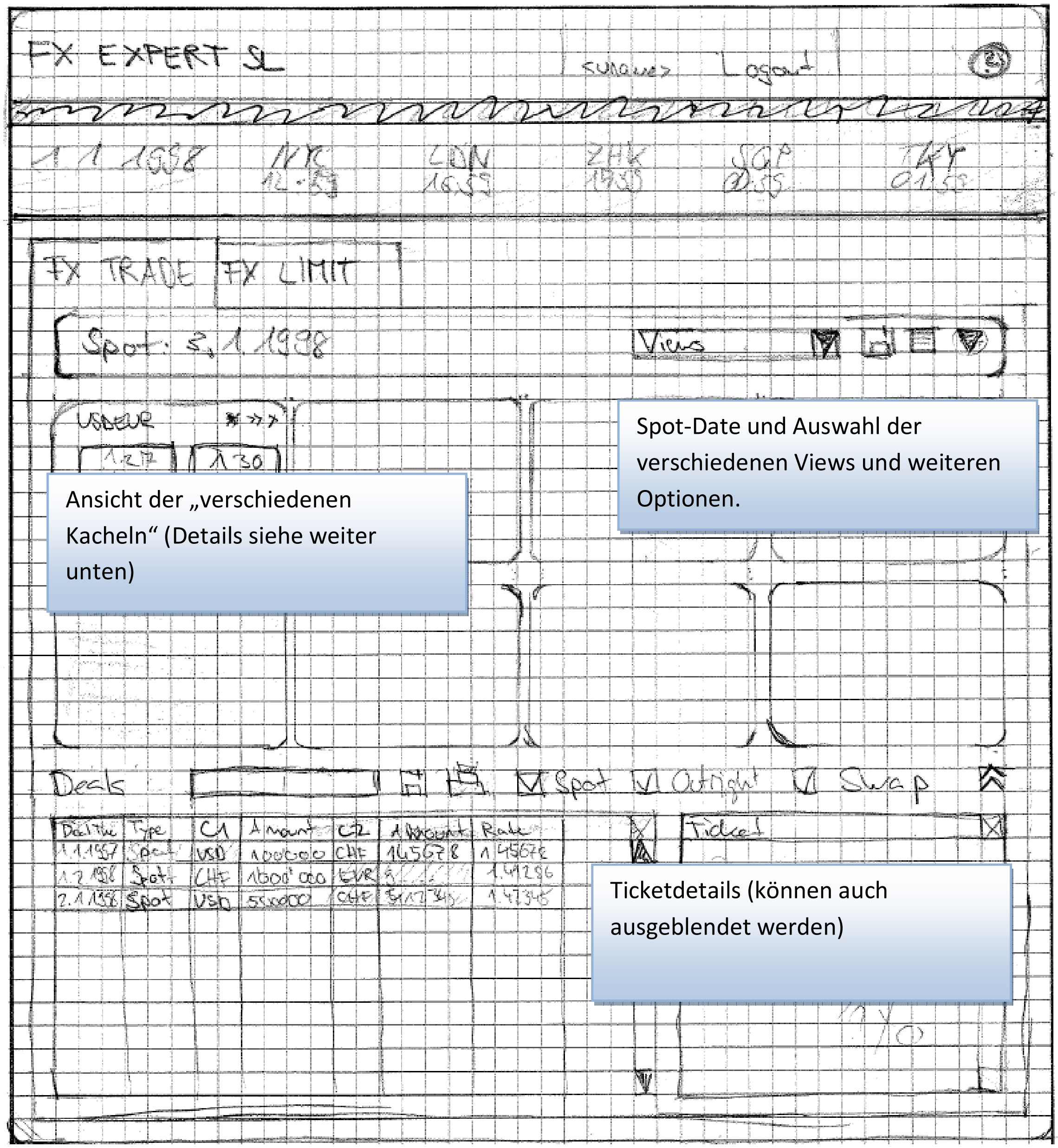
Nr.	*Dokument*	*Autor*
1	Anleitung zur bestehenden Java-Applet Version Manual_FX_Expert.pdf	CS

2 Login View



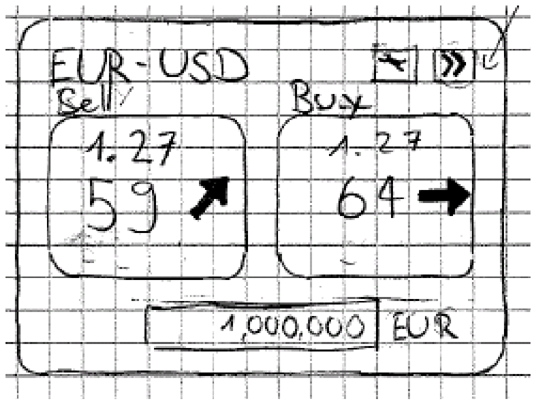
Das Login ist grundsätzlich sehr einfach. Bei einem nicht erfolgreichen Login–Versuch, erscheint eine Fehlermeldung.

3 FX TRADE View



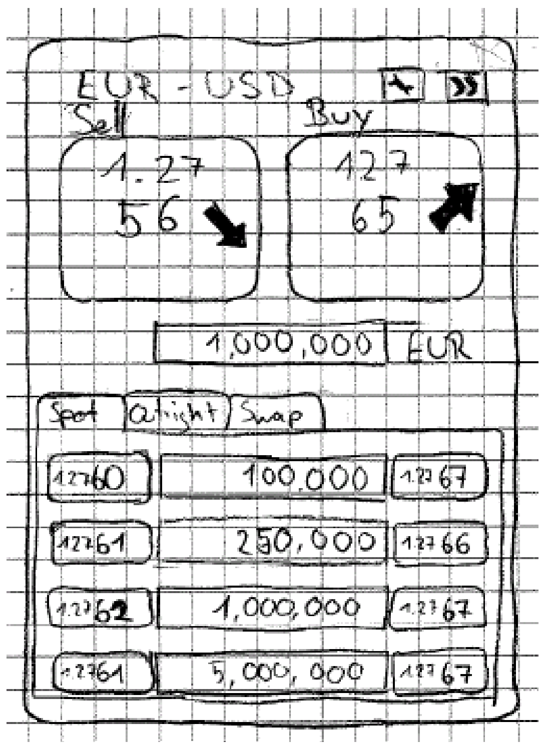
In der FX TRADE View wird das EBS-Chart angezeigt. Man kann verschiedene Ansichten definieren. Die Dealdetails (unten) können auch ausgeblendet, gespeichert und gefiltert werden.

3.1 Tile View



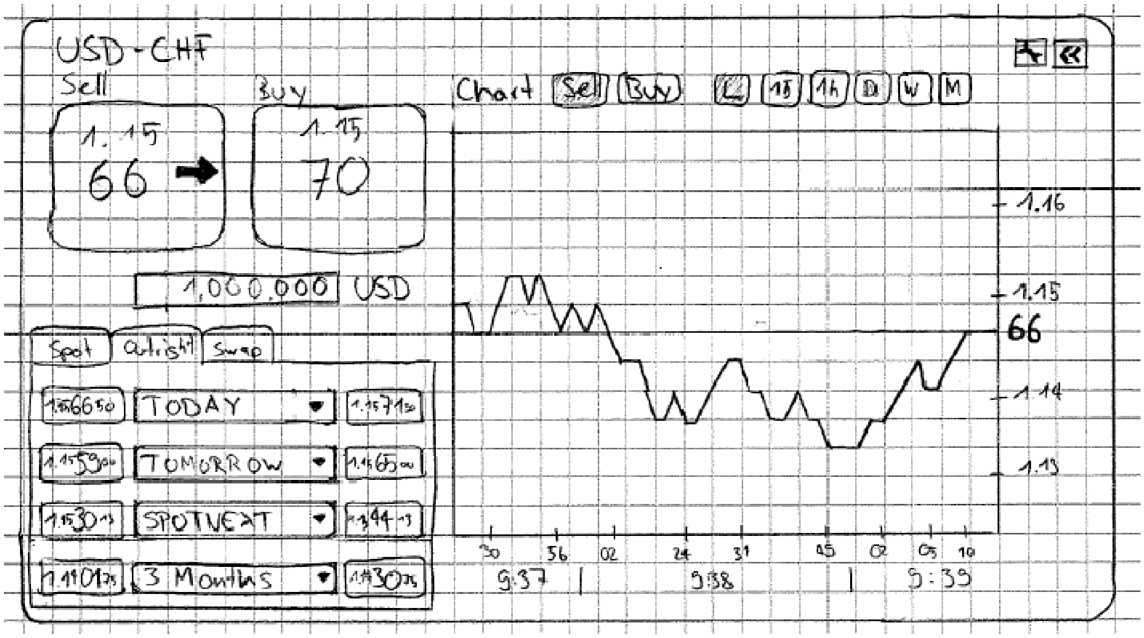
Ansicht mit aktuellem Kurs und direktem Buy/Sell mit Klick auf Button (Hier Sell 59/Buy 64). Die Pfeile geben an, wie sich der Kurs im Vergleich zum letzten Mal verändert hat. Kann vergrössert werden (>>) zur Tile Chart View.

3.2 Tile Detail View



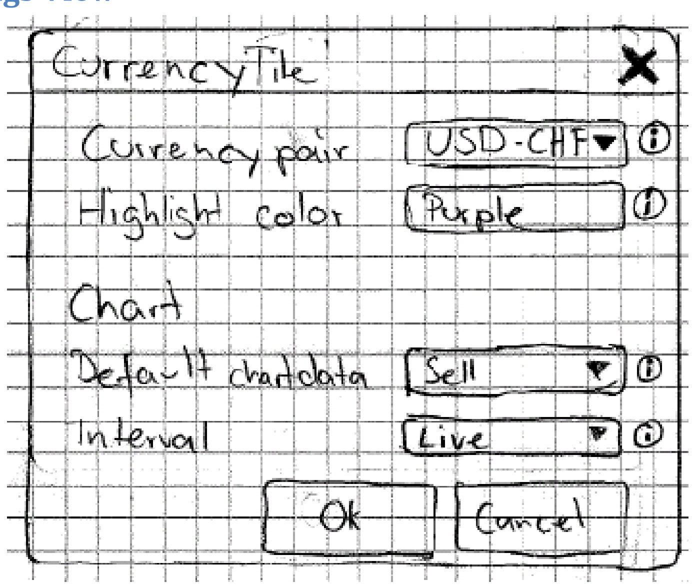
Detailliertere Ansicht der Kachel mit den untertypen Spot, Outright, Swap.

3.3 Tile Chart View



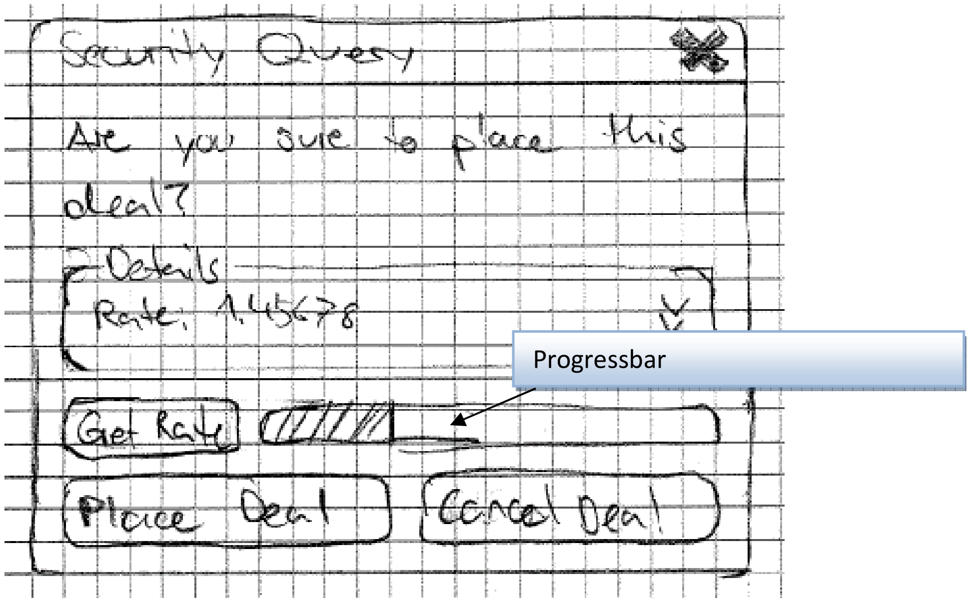
Ansicht der Kachel mit LineChart für eingestelltes Intervall (Live, 15 Minuten, etc.) Ansonsten gleich wie Tile Detail View.

3.4 Tile Settings View



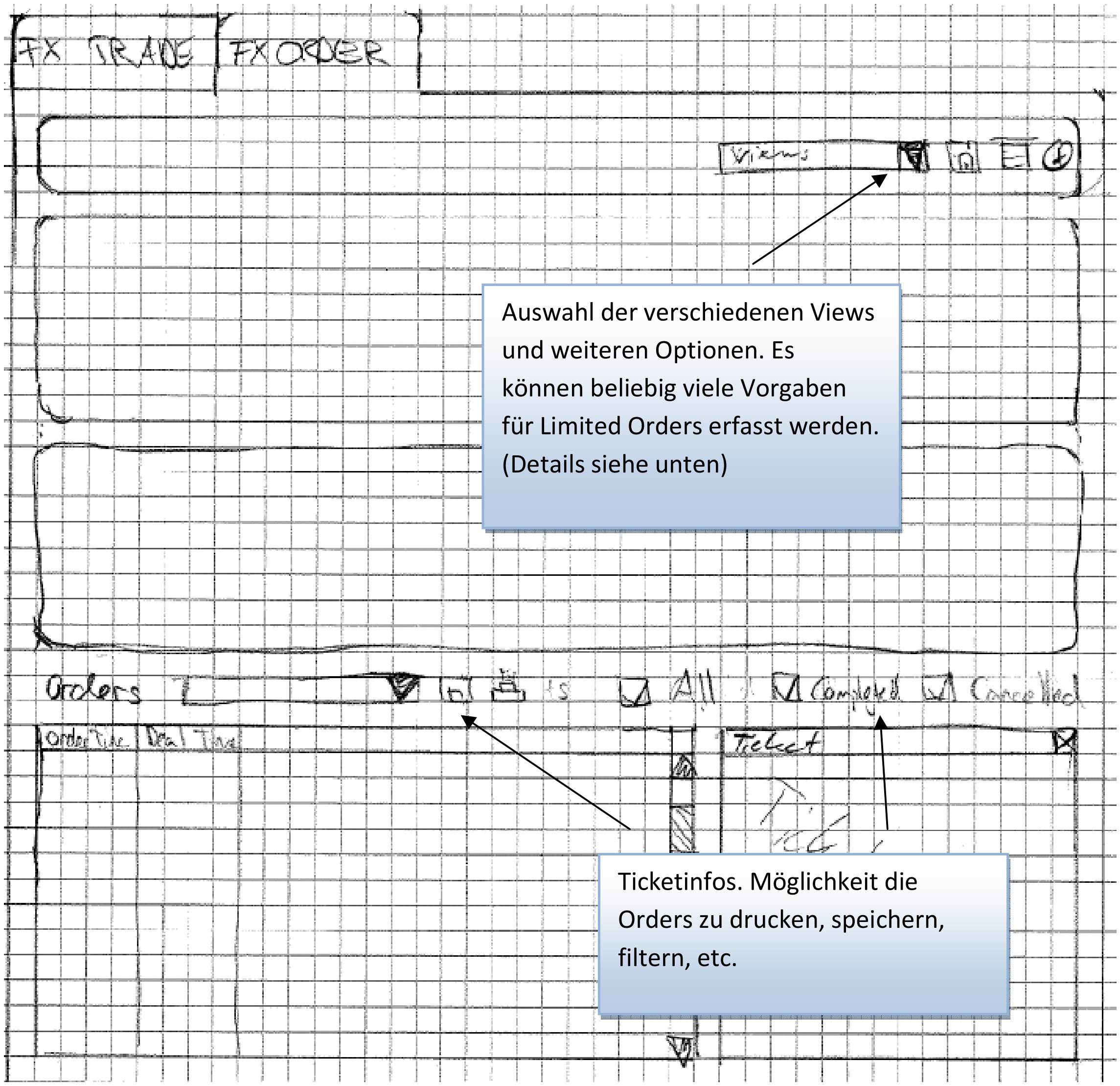
Einstellungen für eine einzelne Kachel. Einstellung des CurrencyPairs, einer Highlight Color und gewissen Charteinstellungen (Standard-Stream, Standard-Intervall)

3.5 Deal Security View



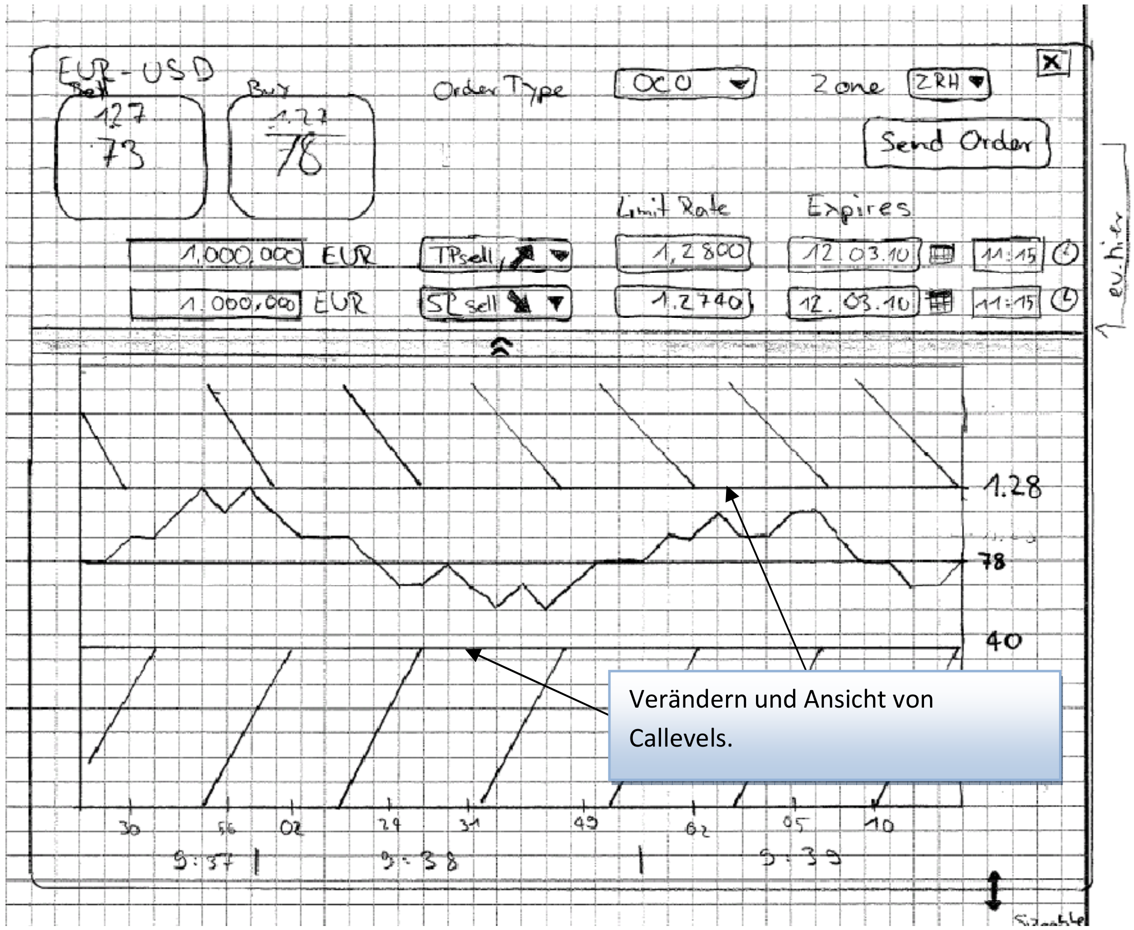
Sicherheitsabfrage bei „Deal" mit ausfahrbaren Details zum Deal und einer Progressbar, die (je nach Zeitvorgabe) hinunterzählt. Beim Ablaufen der Zeit kann die neue Rate via „Get Rate" geholt werden.

4 FX ORDER View



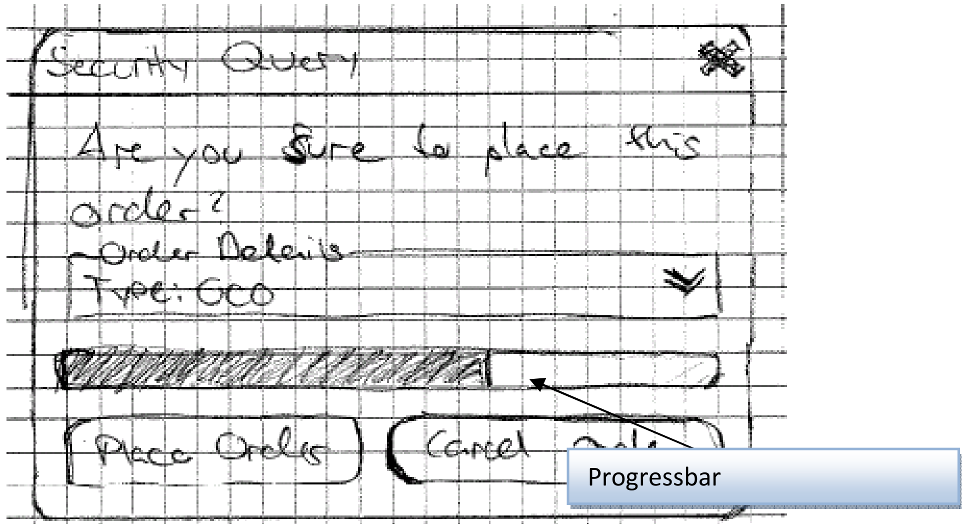
In der FX LIMIT View wird das EBS-Chart angezeigt. Man kann verschiedene Ansichten definieren. Die Orderdetails (unten) können auch ausgeblendet, gespeichert und gefiltert werden.

4.1 OrderDetailView



Detailansicht der Orders. Das LineChart kann ausgeblendet werden. Ordereinstellungen verändern sich je nach Auswahl bei „Order Type“. Im LineChart können Callevels angezeigt und verändert werden. (Hier fehlen noch die Optionen für das Chart (Intervall, Welcher Wert, etc.)

4.2 OrderSecurityCheck



Sicherheitsabfrage bei Order mit ausfahrbaren Details zum Deal und einer Progressbar, welche (je nach Zeitvorgabe) hinunterzählt. Beim Ablaufen der Zeit kann die Order nicht mehr platziert werden.

5 Weitere Views

Hinzukommen noch Ansichten für

- Hilfe
- Benutzereinstellungen
- Evtl. grosse Chart Ansicht

Foreign Exchange UI

Software Architecture Document

1 Dokumentinformationen

1.1 Änderungsgeschichte

Datum	*Version*	*Änderung*	*Autor*
20.10.09	0.1	Dokument erstellt	dh
28.10.09	0.2	Models eingefügt / Update von architektonischer Darstellung	dh
04.11.09	0.3	Review	dh
29.11.09	0.4	Anpassung	dh
10.12.09	0.5	Anpassung / Datenbank hinzugefügt	dh
15.12.09	1.0	Überarbeitung / Review / Verbesserung	ds

1.2 Referenzen

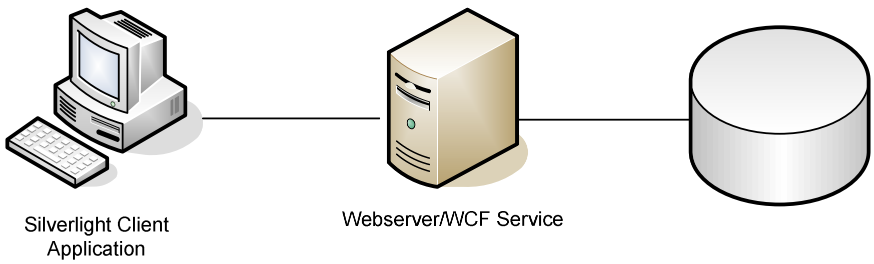
Nr.	*Dokument*	*Autor*
1	DbLINQ: http://code.google.com/p/dblinq2007/	Google
2	Datenbank Provider für LINQ http://en.wikipedia.org/wiki/Language_Integrated_Query#Other_providers	WIKI
3	SL Extensions http://www.slextensions.net/	SL Extensions
4	Silverlight 3 Toolkit July/October 2009 http://www.codeplex.com/Silverlight	Codeplex

2 Architektonische Darstellung

Die Forex UI Silverlight Applikation basiert auf einer Client Server Architektur. Die Silverlight Applikation wird mittels Webserver auf den Client geladen. Das Login geschieht über den WCF Service des Webservers, welcher mit der Datenbank verbunden ist.

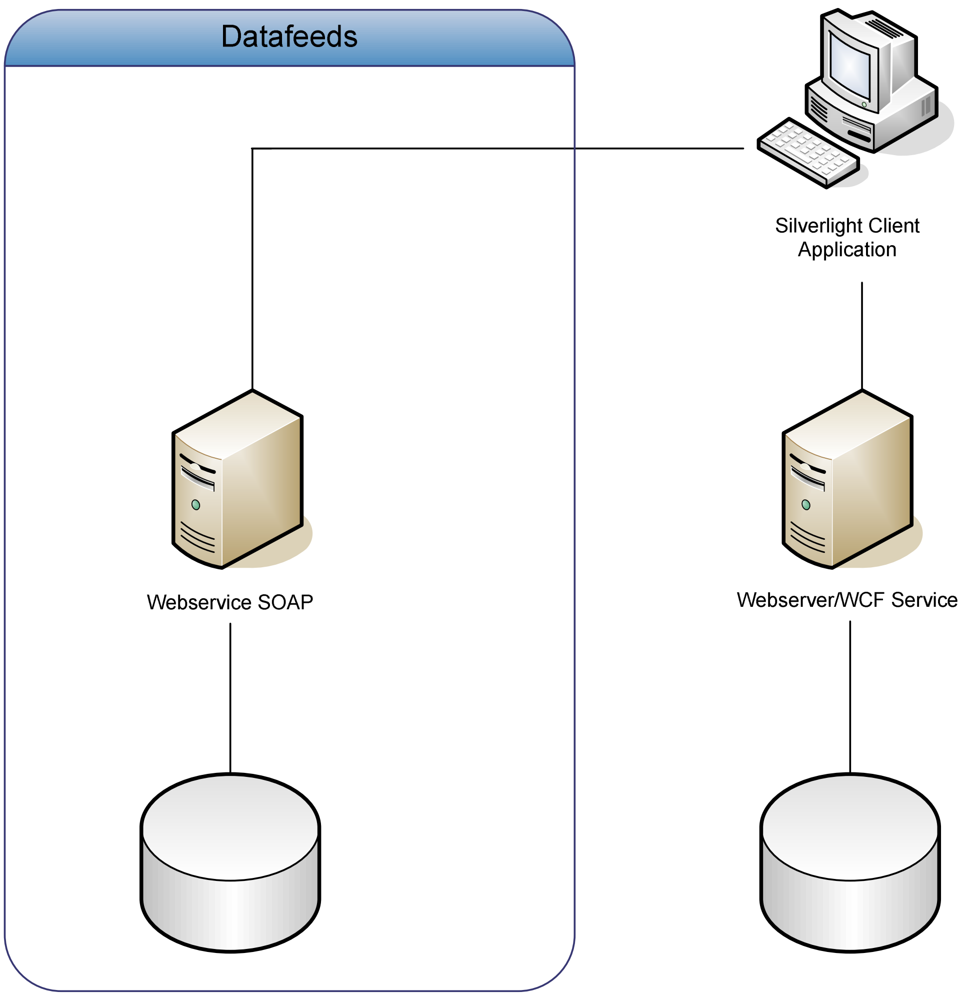
2.1 Mögliches Szenario 1 (Implementiert)

Sämtliche Daten werden über den WCF Service geladen. Um an die Daten zu kommen, muss ein Benutzer authentifiziert sein. Die Forex-Daten sowie die persönlichen Einstellungen werden über den WCF Service gehandhabt.
Es ist auch ohne weiteres möglich, dass der WCF Service z.B. die Forex Daten von einem weiteren Service (z.B. SOAP) herunterlädt und diese so dem Client überspielt.



2.2 Mögliches Szenario 2

Jegliche benutzerspezifischen Daten (Kontoinformationen, Transaktionen) werden über den WCF Service auf Anfrage geladen. Einstellungen, zu welchen auch die EBS Tiles gehören, können über diesen Service gespeichert werden.
Die Live-Datafeeds werden asynchron über den SOAP Webservice konsumiert. Wir gehen davon aus, dass die Datafeeds bankintern gelöst sind. Deswegen simulieren wir die Feeds und schreiben die Daten in die Datenbank.



2.3 Datenzugriff

Der Zugriff auf die Daten erfolgt in unserer Lösung mittels LINQ to SQL auf einen MSSQL Server. Es ist allerdings auch möglich andere Datenbanken zum Beispiel mittels DbLINQ[1] anzusprechen. Eine weitere Liste findet sich im Wikipedia Eintrag zu LINQ[2].

3 Architektonische Ziele und Einschränkungen

3.1 Ziele

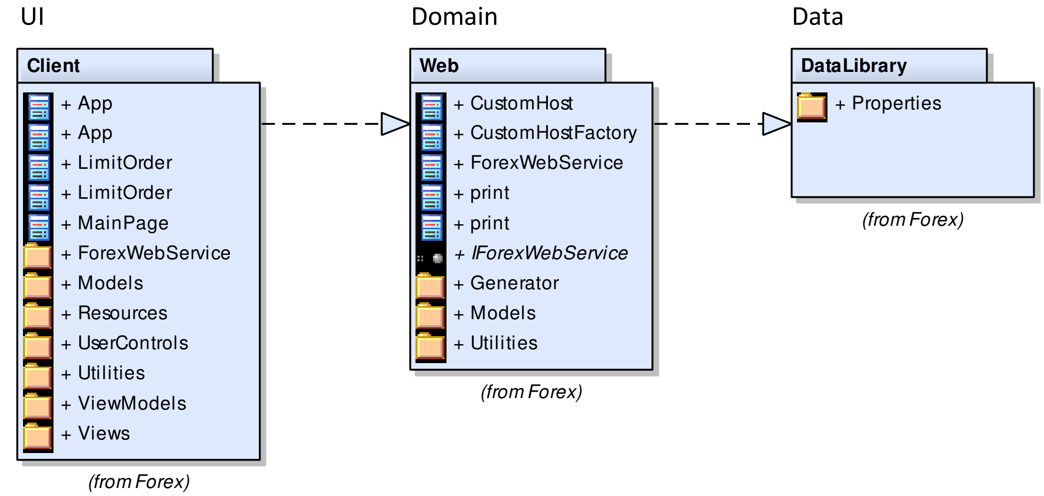
- Die Benutzeroberfläche soll einfach, intuitiv sowie übersichtlich sein. Bei Unklarheiten eines Benutzers, sollte es diesem möglich sein, rasch und möglichst einfach Hilfe für sein Problem zu finden.
- Die Latenzzeit zwischen Client und Server (Datafeed) soll möglichst gering sein.
- Es soll wenn möglich nur eine SSL Session pro Clientanmeldung und nicht für jede Anfrage aufgebaut werden (riesiger Overhead).

3.2 Einschränkungen

- Auf die Sicherheit muss nicht geachtet werden

4 Logische Architektur

4.1 Projekt



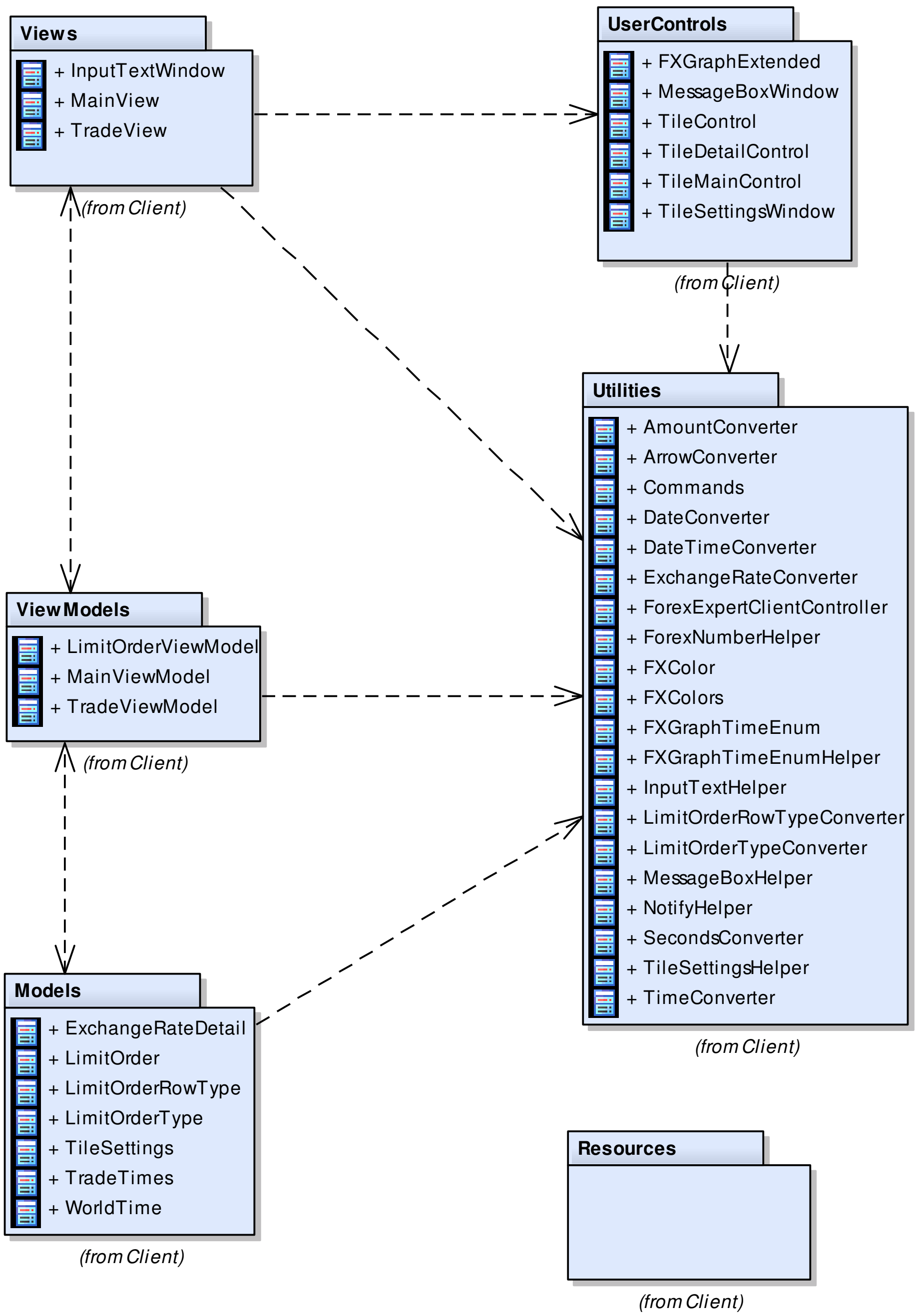
4.2 Forex Client

4.2.1 Libraries

- SLExtensions[3]
  Nötig für die Implementierung des MVVM
  Es gäbe auch noch andere Bibliotheken (MVVM Light, Prism)
- Silverlight 3 Toolkit July/October 2009[4] oder höher
  Für zusätzliche Controls (z.B. WrapPanel)

4.2.2 Aufbau

Das UI ist nach dem MVVM (Model-View-ViewModel) Modell aufgebaut.



4.2.2.1 Views

Ziel ist es in den Views nur die Ansichten über das XAML zu definieren und im Code-Behind eigentlich nichts zu tun. Da Silverlight jedoch noch standardmässig keine Commands im XAML zur Verfügung stellt, werden Events doch noch vom Code-Behind als Commands ans ViewModel weitergesendet. Im Package Views befinden sich auch die UserControls (EBSTile, FXGraph, etc.), welche jedoch zum Teil über mehr Code-Behind verfügen.

4.2.2.2 ViewModels

Das ViewModel besitzt Kenntnis über die View und dem Model und kann also Daten entsprechend weiterreichen. Im Normalfall wird das ViewModel per Data Binding an die View gebunden (zum Teil wegen Einschränkungen von Silverlight auch direkt an die Controls). Daraus entsteht zwar eine enge Kopplung, wobei diese jedoch nicht zwangsweise auch an das Model weitergegeben werden muss.

4.2.2.3 Models

Die Models stellen nun die Daten zur Verfügung. Die Models müssen sich nicht zwingen auf dem Client befinden, sondern können auch die Daten, welche über den WCF/SOAP-Channel kommen, sein.

4.2.2.4 Utilities

Im Package Utilities befinden sich Hilfsklassen wie beispielsweise Converter für das UI (z.B. Umwandeln von DateTime Objekten ins richtige Format).

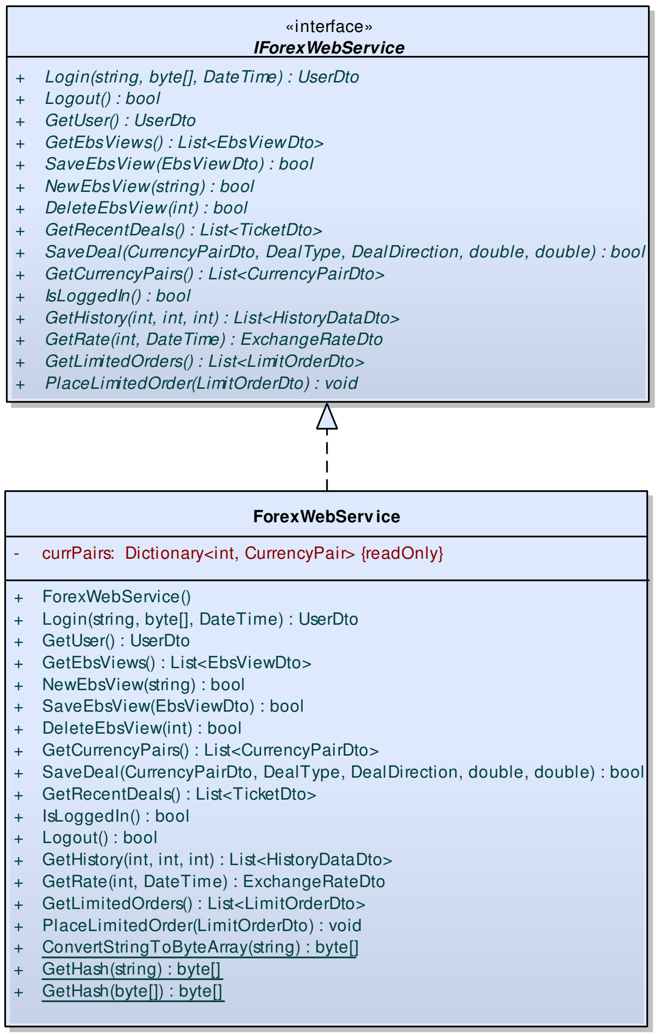
4.2.2.5 Resources

Beinhaltet Grafiken, Texte für die Sprachanpassung, etc.

5 Packages

5.1 Forex Web

5.1.1 Forex Web Service



Die Forex Web Service Klasse implementiert den WCF Service auf dem Webserver. Dieser WCF Service bietet die Schnittstelle zwischen User Interface und Domain.

5.1.1.1 Login

Die Anmeldung erfolgt mittels Benutzername, dem Passwort, welches mit SHA-256 gehasht wird, sowie der Zeit der Anmeldung, damit der Handshake der Authentifikation erfolgreich gehast wird. Zudem legt die Login Methode bei erfolgreicher Anmeldung noch eine Session an, um den User bei zukünftigen Aktivitäten automatisch zu authentisieren. Beim Schliessen der Silverlight Anwendung wird diese vom Webserver automatisch eliminiert.

5.1.1.2 GetUser

Gibt den Benutzer zurück, welcher am System angemeldet ist

5.1.1.3 GetEbsViews

Lädt die EbsViews des angemeldeten Benutzers in eine Liste und gibt diese zurück.

5.1.1.4 GetEbsView

Gibt die angegebene EbsView des angemeldeten Benutzers zurück.

5.1.1.5 NewEbsView

Legt eine neue EbsView mit dem angegebenen Namen an. Falls eine EbsView mit demselben Namen bereits angelegt ist oder der Benutzer nicht authentisiert werden konnte, wird diese nicht gespeichert.

5.1.1.6 SaveEbsView

Speichert die angegebene EbsView des Benutzers und gibt an, ob diese gespeichert werden konnte.

5.1.1.7 DeleteEbsView

Löscht die angegebene EbsView des angemeldeten Benutzers.

5.1.1.8 GetCurrencyPairs

Gibt alle Währungspaare in einer Liste zurück.

5.1.1.9 SaveDeal

Speichert ein Deal (SPOT) des angemeldeten Benutzers.

5.1.1.10 GetRecentDeals

Die letzten getätigten Deals (SPOT) des angemeldeten Benutzers abrufen.

5.1.1.11 IsLoggedIn

Prüfung ob der Benutzer bereits angemeldet ist.

5.1.1.12 Logout

Meldet den Benutzer vom System ab und löscht dessen Session.

5.1.1.13 GetHistory

GetHistory liefert eine Liste von Raten eines bestimmten Währungspaares. Die Liste wird durch ein Intervall gruppiert. Das Intervall kann von einer Minute bis hin zu Stunden sein. Man erhält jeweils die tiefsten sowie höchsten buy/sell-Werte von jeder Zeitspanne.

5.1.1.14 GetRate

Die GetRate Methode liefert nach dem Aufruf eine neue Rate eines Währungspaares zurück. Dies kann, je nachdem wie schnell der Generator die Rate generiert, zwischen einigen hundert Millisekunden bis zu einigen Sekunden dauern.

5.1.1.15 GetLimitedOrders

Gibt die erfassten limitierten Aufträge des angemeldeten Benutzers zurück.

5.1.1.16 PlaceLimitedOrder

Erfasst einen limitierten Order in der Datenbank

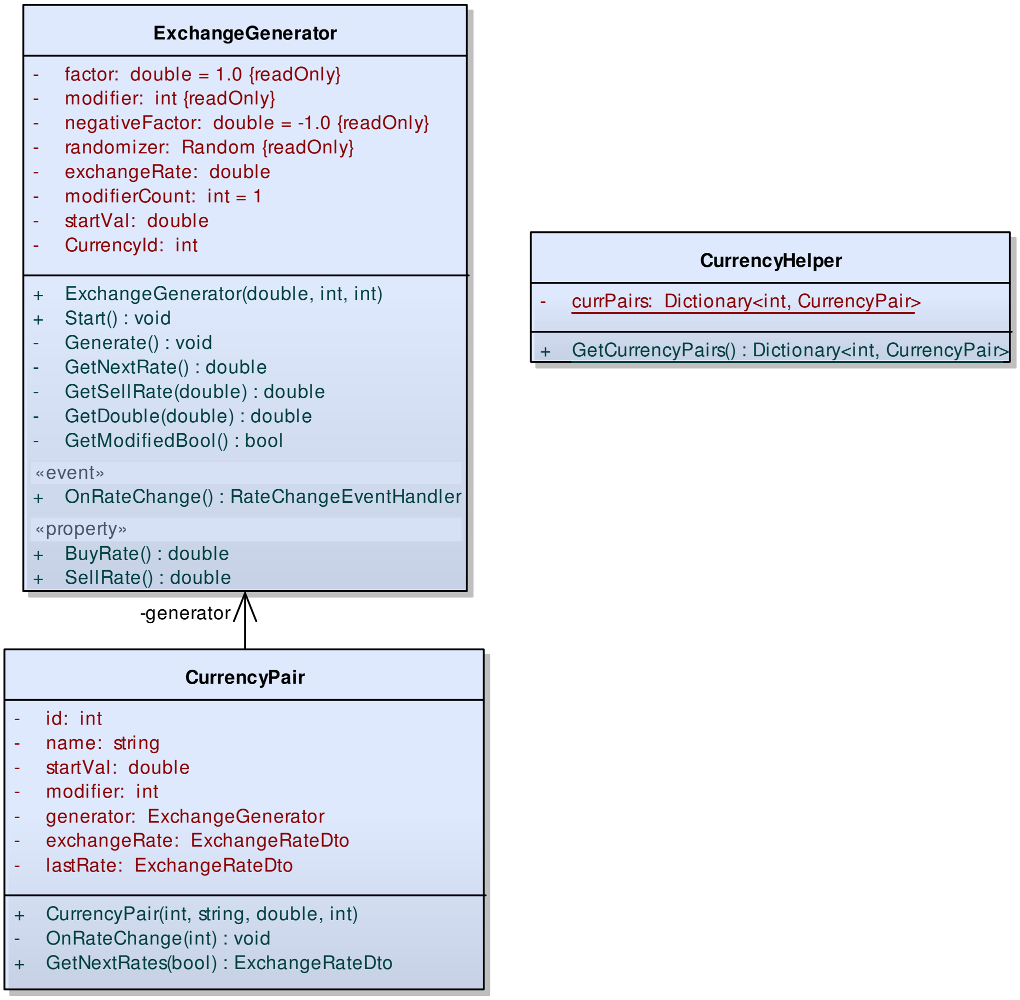
5.1.1.17 ConvertStringToArray

Wandelt einen String in ein Byte-Array um. Dies wird benötigt, um einen String zu hashen.

5.1.1.18 GetHash

Die GetHash Methoden liefern ein mit SHA-256 gehashtes Byte-Array des Eingabewortes/arrays.

5.1.2 Generator



Der Generator ist für die kontinuierliche Erzeugung der fiktiven Forex-Daten zuständig. Jede CurrencyPair Instanz besitzt dazu ihren eigenen ExchangeGenerator. Die Daten werden in einem eigenen Thread erzeugt, damit die Anwendung blockierungsfrei bleibt.

5.1.2.1 ExchangeGenerator

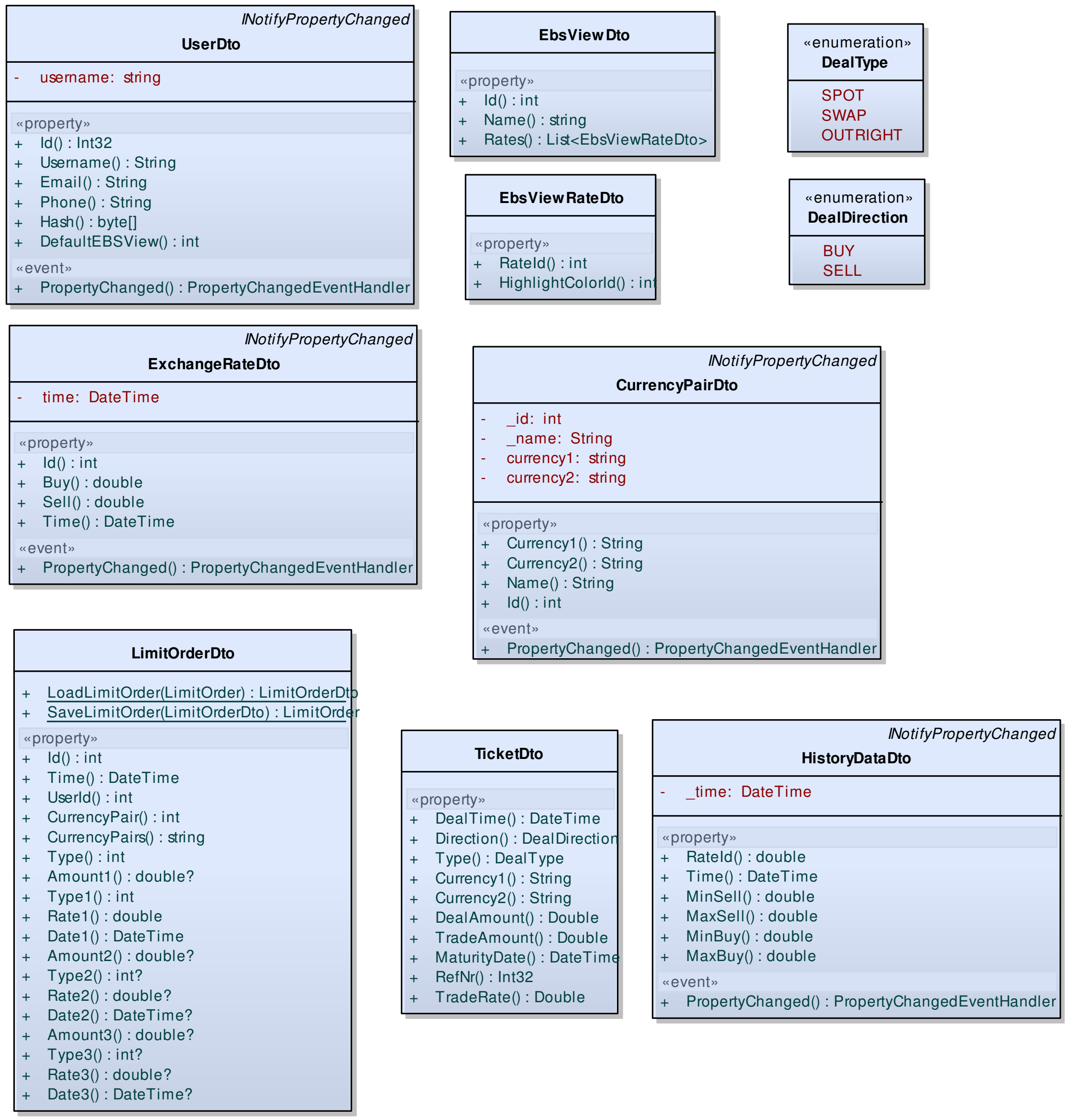
5.1.2.1.1 ExchangeGenerator Konstruktor

Konstruktor mit Parametern für den Initialwert des Währungspaares, Modifizierungswert (um beim Testen z.B. sinkende oder steigende Flanken schnell simulieren zu können) sowie das Währungspaar.

5.1.2.1.2 Start

Startet einen Thread, welcher das Generieren der Raten vornimmt (GetNextRate). Die Raten werden in zufälligen Abständen zwischen mehreren Zehntelsekunden und einigen Sekunden generiert. Dies entspricht dem möglichen Updateintervall beim Fremdwährungshandel.

5.1.3 Models



Die Models-Datenklassen werden für die Übertragung des WCF Services verwendet. Sie bestehen meist nur aus Properties, da sie keine Funktionalität implementieren. Somit wird sichergestellt, dass die übertragenen Daten nicht unnötig Ballast verursachen und der WCF Service alle Datentypen versteht. (z.B. KeyValuePair <K,V> versteht der WCF Service nicht)

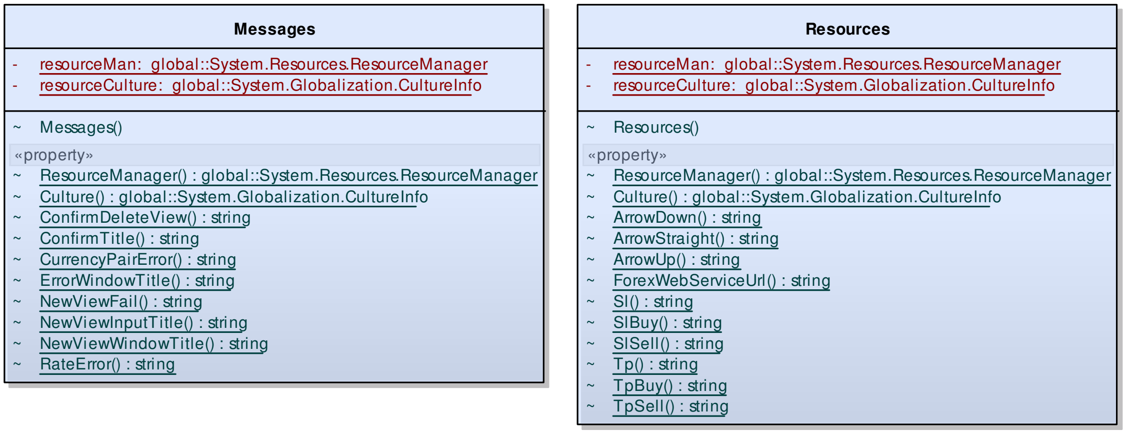
5.1.4 Utilities



Die Utilities sind Hilfsklassen, welche global nutzbare Methoden innerhalb des Webservices anbieten.

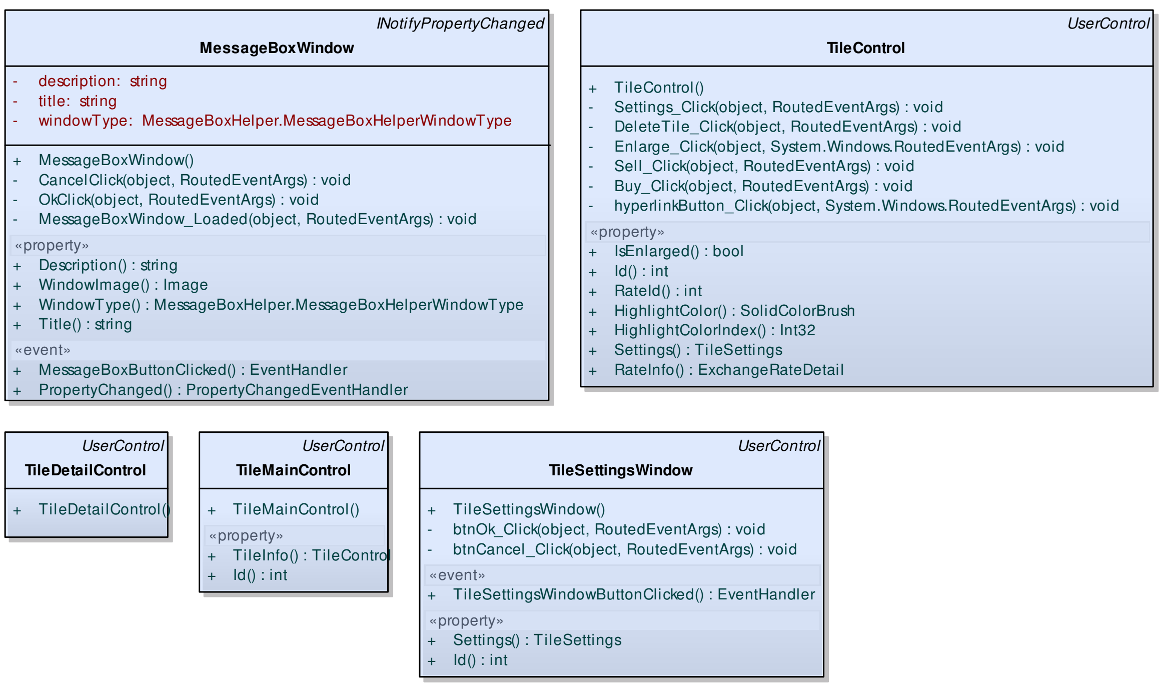
5.2 Forex Client

5.2.1 Resources



Die Ressourcen beinhalten Texte, welche im UI angezeigt werden, um diese schneller und einfacher lokalisieren zu können.

5.2.2 UserControls



Die UserControls sind mehrfach verwendete, grafische Komponenten, welche aus mehreren Unterelementen bestehen.

5.2.2.1 MessageBoxWindow

Das MessageBoxWindow ist eine Messagebox, welche sich beim Aufruf über die gesamte Applikation legt und die restlichen Controls überdeckt (eine Art modaler Dialog).

5.2.2.2 TileControl

Das TileControl bildet den Rahmen eines TileDetailControls.

5.2.2.3 *TileDetailControl*

Das TileDetailControl enthält alle Komponenten eines Tiles wie z.B. Sell/Buy Buttons mit Raten, FxGraphExtended usw. Zudem enthält es die 3 verschiedenen Status Default, Extended, LimitOrder.

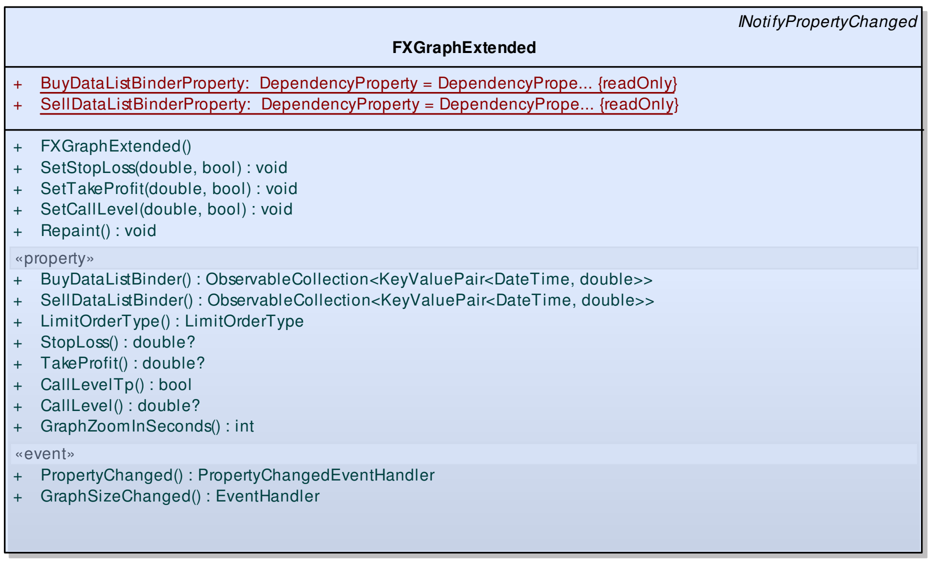
5.2.2.4 *TileMainControl*

Das TileMainControl ist die Hauptansicht eines Tiles.

5.2.2.5 *TileSettingsWindow*

Ist das Einstellungsfenster für die Tiles, auch welchem das Währungspaar sowie die Highlight-Farbe gesetzt werden kann.

5.2.2.6 *FxGraphExtended*



Dieses UserControl zeigt einen Graphen mit Buy/Sell Linie an, welcher mit Listen von ExchangeRates Daten abgefüllt werden kann. Es bietet zudem Funktionalitäten, um die LimitedOrders grafisch erfassen zu können.
In der obigen Grafik sind nur die Public-Member zu sehen.

5.2.2.6.1 SetStopLoss / SetTakeProfit / SetCallLevel

Setzt den Wert der StopLoss/TakeProfit/CallLEvel-Linie, welche bei den LimitedOrders angezeigt werden kann.

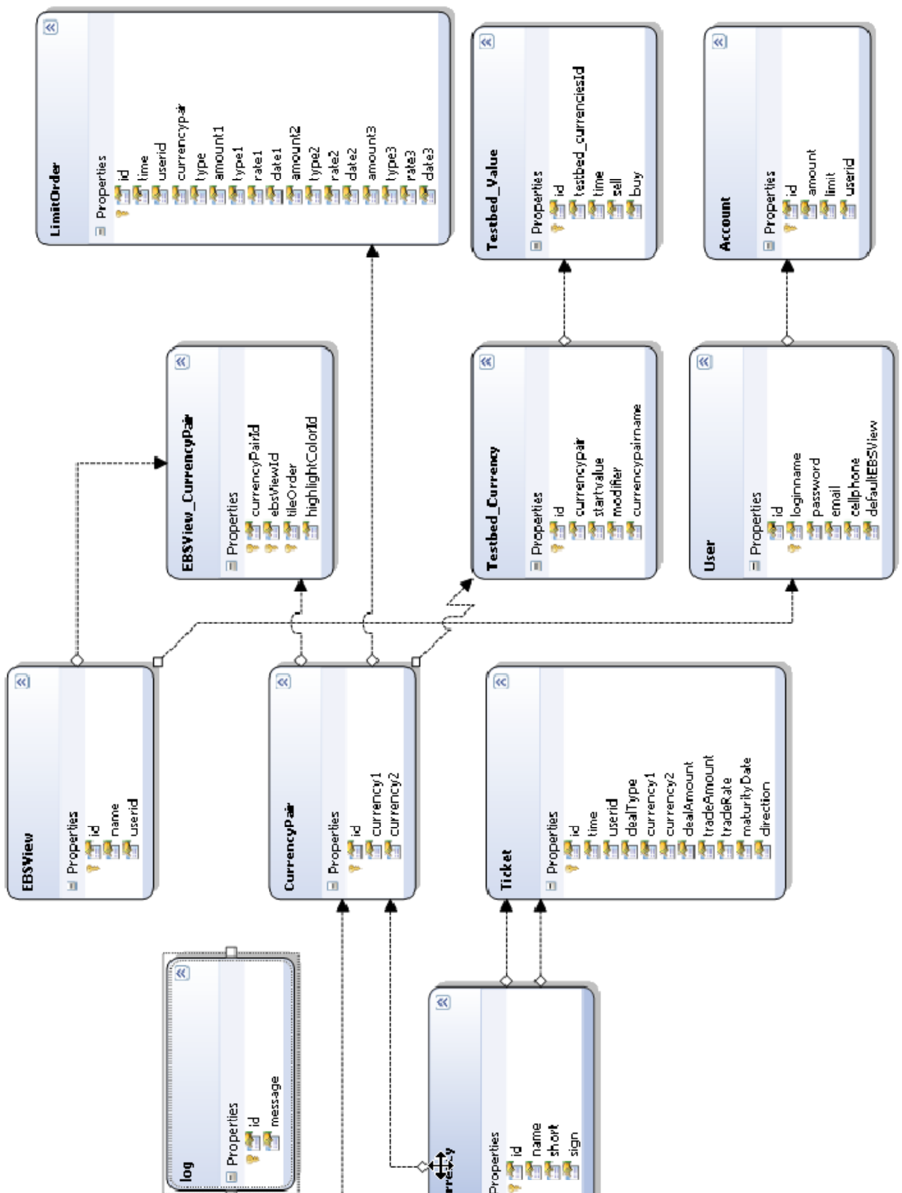
5.2.2.6.2 Repaint

Zeichnet den Graphen mit seinen Daten neu.

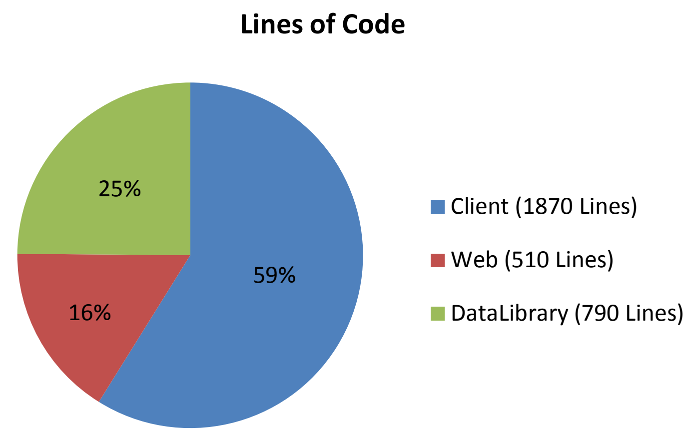
6 Sequenzdiagramm

Das Sequenzdiagramm, welches die Benutzeranmeldung bis hin zur Ratenabfrage darstellt, befindet sich im Anhang.

7 Datenbank

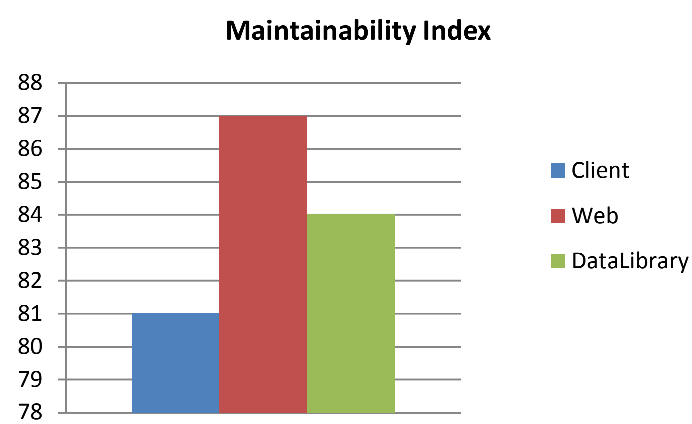


8 Codeauswertung



Der Client hat beinhaltet mit Abstand am meisten Code. Dies ist damit zu erklären, dass die ViewModels einiges an Code zur Implementierung beinhalten.
Die Codezeilen beinhalten keine XAML Dateien.

Der Maintainability-Index gibt an, wie gut welches Projekt wie wart bar ist. Je höher der Score, desto besser die Wartbarkeit.
Der Client ist sicherlich am schwierigsten zu warten, da es auch das komplexeste Projekt ist.



Testprotokolle

1 Test vom 27.11.09

Durchgeführt von Dominik Süsstrunk

1.1 Systemtests

Testfall	**Benutzerlogin**
Beschreibung	Benutzer, welcher in der Datenbank erfasst ist, ans System anmelden.
Erwartetes Resultat	Bei korrekter Eingabe ist der Benutzer am System angemeldet und kann Währungen handeln. Bei einer fehlerhaften Eingabe wir dem Benutzer ein Fehler angezeigt.
Resultat	Verhalten wie beschrieben

1.1.1 Tiles

Testfall	**Hinzufügen eines Tiles**
Beschreibung	Durch klick auf den Tiles-Hinzufügen Button, soll ein Einstellungsfenster geöffnet werden, auf welchem man die Einstellungen des Tiles vornehmen kann. Bestätigt man die Eingaben soll ein Tile der EBS View hinzugefügt werden.
Erwartetes Resultat	Nach Auswahl des Währungspaares auf dem Einstellungsfenster und der Bestätigung des Dialoges muss ein neues Tile in der EBS Ansicht mit dem ausgewählten Währungspaar angezeigt werden. Wird eine HighlightColor ausgewählt, muss das Tile die ausgewählte Farbe annehmen.
Resultat	Währungspaar stimmt jedoch keine HighlightColor
Zusätzliche Information	HighlightColor noch nicht implementiert

Testfall	**Einstellungen eines Tiles ändern**
Beschreibung	Die Einstellungen eines Tiles können zur Laufzeit geändert werden, auch das Währungspaar.
Erwartetes Resultat	Nach Auswahl des Währungspaares auf dem Einstellungsfenster und der Bestätigung des Dialoges muss ein neues Tile in der EBS Ansicht mit dem ausgewählten Währungspaar angezeigt werden. Wird eine Highlight-Color ausgewählt, muss das Tile die ausgewählte Farbe annehmen.
Resultat	Währungspaar ändert sich jedoch nicht die HighlightColorss
Zusätzliche Information	HighlightColor noch nicht implementiert

Testfall	**Tile Position ändern**
Beschreibung	Die Position der einzelnen Tiles kann durch Drag & Drop geändert werden.
Erwartetes Resultat	Nach dem Drag & Drop der Tiles werden diese an der neuen Position angezeigt.
Resultat	Nicht implementiert

Testfall	**Tile vergrössern/verkleinern**

Beschreibung	Wird das Tile vergrössert, so zeigt es einen Graphen des Verlaufs des Währungspaares.
Erwartetes Resultat	Die Tiles können vergrössert/verkleinert werden.
Resultat	Chart mit Verlauf wird angezeigt. Zum Teil noch unsaubere Animation

1.1.2 EBS Views

Testfall	**Laden der EBS Views**
Beschreibung	Ein angemeldeter Benutzer sieht seine angelegten EBS Views und kann zwischen diesen wechseln.
Erwartetes Resultat	Die EBS Views werden in der Liste angezeigt und nach dem Wechseln einer EBS View, werden die entsprechenden Tiles angezeigt. Die DefaultView wird geladen.
Resultat	EBSViews werden in Liste angezeigt. Default View wird nicht immer sauber geladen, z.T. noch alte View

Testfall	**Anlegen einer neuen EBS View**
Beschreibung	Der Benutzer hat die Möglichkeit, neue EBS Views anzulegen. Dazu klickt dieser auf den Knopf ‚neue EBS View anlegen' (Neu-Symbol) und gibt den Namen der View ein. Diese wird nun eröffnet und in der Liste angezeigt.
Erwartetes Resultat	Wird eine neue EBS View angelegt, ist diese in der Datenbank dem entsprechenden Benutzer zugeordnet und wird dem Benutzer angezeigt.
Resultat	Gespeicherte View ist in der Datenbank für den jeweiligen User hinterlegt.

Testfall	**Speichern einer EBS View**
Beschreibung	Die EBS View enthält verschiedene Tiles welche der Benutzer hinzugefügt hat. Diese kann er laufend anpassen (Currency-Pair wechseln, löschen, hinzufügen). Änderungen an einer EBS View werden durch den Speichern Knopf übernommen.
Erwartetes Resultat	Nach dem Speichern einer EBS View wird diese auch beim nächsten Start der Applikation so geladen. Auch nach dem Wechseln der Views muss der gespeicherte Zustand angezeigt werden. Werden die Änderungen nicht gespeichert muss nach dem Wechseln der Views oder nach einer Neuanmeldung der alte, gespeicherte Zustand geladen werden.
Resultat	Änderungen an einer View werden in der Datenbank gespeichert

1.1.3 Buy/Sell

Testfall	**Buy/Sell auf einem Tile**
Beschreibung	Der Benutzer hat die Möglichkeit schnell und einfach Währungen zu kaufen oder verkaufen. Durch einen Klick auf den Buy oder Sell Button eines Tiles wird ein solcher Auftrag erfasst.
Erwartetes Resultat	Nach dem Buy (Sell) Klick, wird von der Währung 1 so viel ‚gekauft (verkauft)' wie eingegeben wurde. Zuerst muss der Benutzer jedoch noch in einem Fenster die Daten bestätigen. Ein Ticket muss dann erfasst sein (angezeigt bei den Tickets)
Resultat	Sicherheitsabfrage fehlt. Rest stimmt

Testfall	**Tickets filtern**
Beschreibung	Ticket können nach Tickettyp (SPOT, SWAP, OUTRIGHT) gefiltert werden

	und nach allen Feldern sotiert werden.
Erwartetes Resultat	Je nachdem welche Typen ausgewählt sind (Checkboxen) sollen nur jeweilige Tickets angezeigt werden. Auch nach einer Aktualisierung soll der Filter weiter bestehen, wie auch die Sortierung
Resultat	Einträge können gefiltert und sortiert werden, sind jedoch nach einer Aktualisierung wieder zurückgesetzt.

Testfall	**Ticketinfo anzeigen**
Beschreibung	Anzeigen von detaillierteren Informationen zu einem Ticket
Erwartetes Resultat	Nach Auswahl eines Tickets werden die Inforamtionen zu diesem Ticket angezeigt.
Resultat	Feld wird zwar angezeigt, aber ohne Inhalt

1.1.4 FX Graph / Limited Order

Testfall	**Graph anzeige wechseln**
Beschreibung	Der FX Graph unterstützt mehrere Ansichtsarten. Die Buy/Sell Linien können ein/ausgeblendet werden und die Zeitachse ist durch einen Slider sowie das Mausrad wechselbar.
Erwartetes Resultat	Wird der Sell/Buy Knopf aktiviert, so müssen die entsprechenden Linien im Graph angezeigt werden. Sind beide Knöpfe deaktiviert, so wird automatisch die Buy Linie angezeigt. Die Zeitachse muss durch das Mausrad sowie den Slider gewechselt werden können und die Beschriftung der Zeitachse muss stimmen.
Resultat	Verhalten ist wie beschrieben.

Testfall	**Limit Order Modus aktivieren**
Beschreibung	Oberhalb des Graphen befindet sich ein Link um den Limit Order Modus zu aktivieren. Dieser blendet die Limit Order Leiste ein respektive aus.
Erwartetes Resultat	Wird der Link angeklickt muss die Limit Order Leiste angezeigt oder ausgeblendet werden.
Resultat	Limit Order Modus wird aktiviert

Testfall	**Limit Order Liste / FX Graph**
Beschreibung	Der FX Graph kann mit der Limit Order Liste zusammenarbeiten und zeigt Stop Loss / Take Profit sowie Call Levels. Die Eingaben in die Limit Order Liste müssen in den FX Graph übertragen werden. Eingaben in dem Graph (Ctrl – Mausklick und Rauf-/Runterbewegen der Maus) müssen in der Liste angezeigt werden
Erwartetes Resultat	Wird ein Call Level eingegeben, wird dieser mittels blauer Linie mit Verlauf angezeigt (Take Profit, Verlauf nach oben, Stop Loss Verlauf nach unten). Wird ein Stop Loss oder Take Profit eingegeben, so werden diese rot/grün dargestellt. Eingaben von Stop Loss / Take Profit mit der Maus müssen in der Liste angezeigt werden. Diese werden durch CTRL - Klick + Mausbewegung rauf-/runter gezeichnet. Eingabe des Call Levels mittels CTRL – Doppelklick (nur bei Typ Call Level möglich).
Resultat	Verhalten funktioniert noch nicht wirklich, muss noch fertig gestellt werden.

Testfall	**Limit Order speichern**
Beschreibung	Einen Limit Order soll man mittels klick auf den Send Limit Order Button speichern können. Die getätigten Eingaben werden dann gespeichert und würden vom Banksystem ausgewertet werden.
Erwartetes Resultat	Nach dem senden eines Limit Orders wird ein Fenster angezeigt, auf welchem man den Limit Order noch einmal bestätigen muss. Bestätigt man diesen so soll ein Datensatz in der Datenbank angelegt sein und dieser muss in der Limit Order Ticket-Liste erscheinen. Das Eingabeformular des Limit Orders wird daraufhin geleert um einen neuen Limit Order zu erfassen.
Resultat	Nicht Implementiert

2 Test vom 04.12.09

Durchgeführt von Daniel Häfliger

2.1 Systemtests

Testfall	Benutzerlogin
Beschreibung	Benutzer, welcher in der Datenbank erfasst ist, ans System anmelden.
Erwartetes Resultat	Bei korrekter Eingabe ist der Benutzer am System angemeldet und kann Währungen handeln. Bei einer fehlerhaften Eingabe wir dem Benutzer ein Fehler angezeigt.
Resultat	Resultat wie erwartet

2.1.1 Tiles

Testfall	Hinzufügen eines Tiles
Beschreibung	Durch klick auf den Tiles-Hinzufügen Button, soll ein Einstellungsfenster geöffnet werden, auf welchem man die Einstellungen des Tiles vornehmen kann. Bestätigt man die Eingaben soll ein Tile der EBS View hinzugefügt werden.
Erwartetes Resultat	Nach Auswahl des Währungspaares auf dem Einstellungsfenster und der Bestätigung des Dialoges muss ein neues Tile in der EBS Ansicht mit dem ausgewählten Währungspaar angezeigt werden. Wird eine Highlight-Color ausgewählt, muss das Tile die ausgewählte Farbe annehmen.
Resultat	Tile wird hinzugefügt und die Highlight-Color sowie das Währungspaar wird gesetzt

Testfall	Einstellungen eines Tiles ändern
Beschreibung	Die Einstellungen eines Tiles können zur Laufzeit geändert werden, auch das Währungspaar.
Erwartetes Resultat	Nach Auswahl des Währungspaares auf dem Einstellungsfenster und der Bestätigung des Dialoges muss ein neues Tile in der EBS Ansicht mit dem ausgewählten Währungspaar angezeigt werden. Wird eine Highlight-Color ausgewählt, muss das Tile die ausgewählte Farbe annehmen.
Resultat	Das Übernehmen der Einstellungen funktioniert

Testfall	Tile Position ändern
Beschreibung	Die Position der einzelnen Tiles kann durch Drag & Drop geändert werden.
Erwartetes Resultat	Nach dem Drag & Drop der Tiles werden diese an der neuen Position angezeigt.
Resultat	Nicht implementiert

Testfall	Tile vergrössern/verkleinern
Beschreibung	Wird das Tile vergrössert, so zeigt es einen Graphen des Verlaufs des Währungspaares.
Erwartetes Resultat	Die Tiles können vergrössert/verkleinert werden.
Resultat	Graph sowie Verlauf werden angezeigt

2.1.2 EBS Views

Testfall	**Laden der EBS Views**
Beschreibung	Ein angemeldeter Benutzer sieht seine angelegten EBS Views und kann zwischen diesen wechseln.
Erwartetes Resultat	Die EBS Views werden in der Liste angezeigt und nach dem Wechseln einer EBS View, werden die entsprechenden Tiles angezeigt. Die DefaultView wird geladen.
Resultat	EBS View wird korrekt geladen und die entsprechenden Tiles werden angezeigt.
Zusätzliche Information	Teilweise langsam beim Start

Testfall	**Anlegen einer neuen EBS View**
Beschreibung	Der Benutzer hat die Möglichkeit, neue EBS Views anzulegen. Dazu klickt dieser auf den Knopf ‚neue EBS View anlegen' (Neu-Symbol) und gibt den Namen der View ein. Diese wird nun eröffnet und in der Liste angezeigt.
Erwartetes Resultat	Wird eine neue EBS View angelegt, ist diese in der Datenbank dem entsprechenden Benutzer zugeordnet und wird dem Benutzer angezeigt.
Resultat	View ist in der Datenbank vorhanden

Testfall	**Speichern einer EBS View**
Beschreibung	Die EBS View enthält verschiedene Tiles welche der Benutzer hinzugefügt hat. Diese kann er laufend anpassen (Currency-Pair wechseln, löschen, hinzufügen). Änderungen an einer EBS View werden durch den Speichern Knopf übernommen.
Erwartetes Resultat	Nach dem Speichern einer EBS View wird diese auch beim nächsten Start der Applikation so geladen. Auch nach dem Wechseln der Views muss der gespeicherte Zustand angezeigt werden. Werden die Änderungen nicht gespeichert muss nach dem Wechseln der Views oder nach einer Neuanmeldung der alte, gespeicherte Zustand geladen werden.
Resultat	Änderungen der View werden in die Datenbank übernommen

2.1.3 Buy/Sell

Testfall	**Buy/Sell auf einem Tile**
Beschreibung	Der Benutzer hat die Möglichkeit schnell und einfach Währungen zu kaufen oder verkaufen. Durch einen Klick auf den Buy oder Sell Button eines Tiles wird ein solcher Auftrag erfasst.
Erwartetes Resultat	Nach dem Buy (Sell) Klick, wird von der Währung 1 so viel ‚gekauft (verkauft)' wie eingegeben wurde. Zuerst muss der Benutzer jedoch noch in einem Fenster die Daten bestätigen. Ein Ticket muss dann erfasst sein (angezeigt bei den Tickets)
Resultat	Sicherheitsabfrage nicht implementiert. Buy und Sell funktionieren

Testfall	**Tickets filtern**
Beschreibung	Ticket können nach Tickettyp (SPOT, SWAP, OUTRIGHT) gefiltert werden und nach allen Feldern sotiert werden.
Erwartetes Resultat	Je nachdem welche Typen ausgewählt sind (Checkboxen) sollen nur

	jeweilige Tickets angezeigt werden. Auch nach einer Aktualisierung soll der Filter weiter bestehen, wie auch die Sortierung
Resultat	Daten werden gefiltert angezeigt, der Filter besteht auch nach einer Aktualisierung, die Sortierung klappt nicht.

Testfall	**Ticketinfo anzeigen**
Beschreibung	Anzeigen von detaillierteren Informationen zu einem Ticket
Erwartetes Resultat	Nach Auswahl eines Tickets werden die Informationen zu diesem Ticket angezeigt.
Resultat	Informationen zu ausgewähltem Ticket wird angezeigt

2.1.4 FX Graph / Limited Order

Testfall	**Graph anzeige wechseln**
Beschreibung	Der FX Graph unterstützt mehrere Ansichtsarten. Die Buy/Sell Linien können ein/ausgeblendet werden und die Zeitachse ist durch einen Slider sowie das Mausrad wechselbar.
Erwartetes Resultat	Wird der Sell/Buy Knopf aktiviert, so müssen die entsprechenden Linien im Graph angezeigt werden. Sind beide Knöpfe deaktiviert, so wird automatisch die Buy Linie angezeigt. Die Zeitachse muss durch das Mausrad sowie den Slider gewechselt werden können und die Beschriftung der Zeitachse muss stimmen.
Resultat	Linien werden ein/ausgeblendet und Slider/Mausrad setzt den Zoom.

Testfall	**Limit Order Modus aktivieren**
Beschreibung	Oberhalb des Graphen befindet sich ein Link um den Limit Order Modus zu aktivieren. Dieser blendet die Limit Order Leiste ein respektive aus.
Erwartetes Resultat	Wird der Link angeklickt muss die Limit Order Leiste angezeigt oder ausgeblendet werden.
Resultat	Limit Order Modus wird angezeigt, allerdings ohne Funktion
Zusätzliche Information	Limit Order erst grafisch implementiert

Testfall	**Limit Order Liste / FX Graph**
Beschreibung	Der FX Graph kann mit der Limit Order Liste zusammenarbeiten und zeigt Stop Loss / Take Profit sowie Call Levels. Die Eingaben in die Limit Order Liste müssen in den FX Graph übertragen werden. Eingaben in dem Graph (Ctrl – Mausklick und Rauf-/Runter bewegen der Maus) müssen in der Liste angezeigt werden
Erwartetes Resultat	Wird ein Call Level eingegeben, wird dieser mittels blauer Linie mit Verlauf angezeigt (Take Profit, Verlauf nach oben, Stop Loss Verlauf nach unten). Wird ein Stop Loss oder Take Profit eingegeben, so werden diese rot/grün dargestellt. Eingaben von Stop Loss / Take Profit mit der Maus müssen in der Liste angezeigt werden. Diese werden durch CTRL - Klick + Mausbewegung rauf-/runter gezeichnet. Eingabe des Call Levels mittels CTRL – Doppelklick (nur bei Typ Call Level möglich).
Resultat	Call-Level kann nicht eingegeben werden, Stop Loss und Take Profit nur in

	Graph eingetragen, nicht in Liste
Zusätzliche Information	Limit Order sowie Call-Level noch nicht implementiert

Testfall	**Limit Order speichern**
Beschreibung	Einen Limit Order soll man mittels klick auf den Send Limit Order Button speichern können. Die getätigten Eingaben werden dann gespeichert und würden vom Banksystem ausgewertet werden.
Erwartetes Resultat	Nach dem senden eines Limit Orders wird ein Fenster angezeigt, auf welchem man den Limit Order noch einmal bestätigen muss. Bestätigt man diesen so soll ein Datensatz in der Datenbank angelegt sein und dieser muss in der Limit Order Ticket-Liste erscheinen. Das Eingabeformular des Limit Orders wird daraufhin geleert um einen neuen Limit Order zu erfassen.
Resultat	Noch nicht Implementiert

3 Test vom 16.12.09

Test durchgeführt von Dominik Süsstrunk

3.1 Systemtests

Testfall	**Benutzerlogin**
Beschreibung	Benutzer, welcher in der Datenbank erfasst ist, ans System anmelden.
Erwartetes Resultat	Bei korrekter Eingabe ist der Benutzer am System angemeldet und kann Währungen handeln. Bei einer fehlerhaften Eingabe wir dem Benutzer ein Fehler angezeigt.
Resultat	Verhalten wie beschrieben

3.1.1 Tiles

Testfall	**Hinzufügen eines Tiles**
Beschreibung	Durch klick auf den Tiles-Hinzufügen Button, soll ein Einstellungsfenster geöffnet werden, auf welchem man die Einstellungen des Tiles vornehmen kann. Bestätigt man die Eingaben soll ein Tile der EBS View hinzugefügt werden.
Erwartetes Resultat	Nach Auswahl des Währungspaares auf dem Einstellungsfenster und der Bestätigung des Dialoges muss ein neues Tile in der EBS Ansicht mit dem ausgewählten Währungspaar angezeigt werden. Wird eine Highlight-Color ausgewählt, muss das Tile die ausgewählte Farbe annehmen.
Resultat	Beides wird korrekt übernommen

Testfall	**Einstellungen eines Tiles ändern**
Beschreibung	Die Einstellungen eines Tiles können zur Laufzeit geändert werden, auch das Währungspaar.
Erwartetes Resultat	Nach Auswahl des Währungspaares auf dem Einstellungsfenster und der Bestätigung des Dialoges muss ein neues Tile in der EBS Ansicht mit dem ausgewählten Währungspaar angezeigt werden. Wird eine Highlight-Color ausgewählt, muss das Tile die ausgewählte Farbe annehmen.
Resultat	Beides wird korrekt übernommen

Testfall	**Tile Position ändern**
Beschreibung	Die Position der einzelnen Tiles kann durch Drag & Drop geändert werden.
Erwartetes Resultat	Nach dem Drag & Drop der Tiles werden diese an der neuen Position angezeigt.
Resultat	Nicht implementiert

Testfall	**Tile vergrössern/verkleinern**
Beschreibung	Wird das Tile vergrössert, so zeigt es einen Graphen des Verlaufs des Währungspaares.
Erwartetes Resultat	Die Tiles können vergrössert/verkleinert werden.
Resultat	Chart mit Verlauf wird angezeigt.

3.1.2 EBS Views

Testfall	**Laden der EBS Views**
Beschreibung	Ein angemeldeter Benutzer sieht seine angelegten EBS Views und kann

	zwischen diesen wechseln.
Erwartetes Resultat	Die EBS Views werden in der Liste angezeigt und nach dem Wechseln einer EBS View, werden die entsprechenden Tiles angezeigt. Die DefaultView wird geladen.
Resultat	EBSViews werden in Liste angezeigt. Default View wird geladen

Testfall	**Anlegen einer neuen EBS View**
Beschreibung	Der Benutzer hat die Möglichkeit, neue EBS Views anzulegen. Dazu klickt dieser auf den Knopf ‚neue EBS View anlegen' (Neu-Symbol) und gibt den Namen der View ein. Diese wird nun eröffnet und in der Liste angezeigt.
Erwartetes Resultat	Wird eine neue EBS View angelegt, ist diese in der Datenbank dem entsprechenden Benutzer zugeordnet und wird dem Benutzer angezeigt.
Resultat	Gespeicherte View ist in der Datenbank für den jeweiligen User hinterlegt.

Testfall	**Speichern einer EBS View**
Beschreibung	Die EBS View enthält verschiedene Tiles welche der Benutzer hinzugefügt hat. Diese kann er laufend anpassen (Currency-Pair wechseln, löschen, hinzufügen). Änderungen an einer EBS View werden durch den Speichern Knopf übernommen.
Erwartetes Resultat	Nach dem Speichern einer EBS View wird diese auch beim nächsten Start der Applikation so geladen. Auch nach dem Wechseln der Views muss der gespeicherte Zustand angezeigt werden. Werden die Änderungen nicht gespeichert muss nach dem Wechseln der Views oder nach einer Neuanmeldung der alte, gespeicherte Zustand geladen werden.
Resultat	Änderungen an einer View werden in der Datenbank gespeichert

3.1.3 Buy/Sell

Testfall	**Buy/Sell auf einem Tile**
Beschreibung	Der Benutzer hat die Möglichkeit schnell und einfach Währungen zu kaufen oder verkaufen. Durch einen Klick auf den Buy oder Sell Button eines Tiles wird ein solcher Auftrag erfasst.
Erwartetes Resultat	Nach dem Buy (Sell) Klick, wird von der Währung 1 so viel ‚gekauft (verkauft)' wie eingegeben wurde. Zuerst muss der Benutzer jedoch noch in einem Fenster die Daten bestätigen. Ein Ticket muss dann erfasst sein (angezeigt bei den Tickets)
Resultat	Sicherheitsabfrage fehlt. Rest stimmt

Testfall	**Tickets filtern**
Beschreibung	Ticket können nach Tickettyp (SPOT, SWAP, OUTRIGHT) gefiltert werden und nach allen Feldern sotiert werden.
Erwartetes Resultat	Je nachdem welche Typen ausgewählt sind (Checkboxen) sollen nur jeweilige Tickets angezeigt werden. Auch nach einer Aktualisierung soll der Filter weiter bestehen, wie auch die Sortierung
Resultat	Einträge können gefiltert und sortiert werden. Nach Änderung ist Filter noch aktiv, nicht aber die Sortierung

Testfall	**Ticketinfo anzeigen**
Beschreibung	Anzeigen von detaillierteren Informationen zu einem Ticket

Erwartetes Resultat	Nach Auswahl eines Tickets werden die Informationen zu diesem Ticket angezeigt.
Resultat	Feld wird angezeigt mit korrektem Inhalt

3.1.4 FX Graph / Limited Order

Testfall	**Graph anzeige wechseln**
Beschreibung	Der FX Graph unterstützt mehrere Ansichtsarten. Die Buy/Sell Linien können ein/ausgeblendet werden und die Zeitachse ist durch einen Slider sowie das Mausrad wechselbar.
Erwartetes Resultat	Wird der Sell/Buy Knopf aktiviert, so müssen die entsprechenden Linien im Graph angezeigt werden. Sind beide Knöpfe deaktiviert, so wird automatisch die Buy Linie angezeigt. Die Zeitachse muss durch das Mausrad sowie den Slider gewechselt werden können und die Beschriftung der Zeitachse muss stimmen.
Resultat	Verhalten ist wie beschrieben.

Testfall	**Limit Order Modus aktivieren**
Beschreibung	Oberhalb des Graphen befindet sich ein Link um den Limit Order Modus zu aktivieren. Dieser blendet die Limit Order Leiste ein respektive aus.
Erwartetes Resultat	Wird der Link angeklickt muss die Limit Order Leiste angezeigt oder ausgeblendet werden.
Resultat	Limit Order Modus wird aktiviert

Testfall	**Limit Order Liste / FX Graph**
Beschreibung	Der FX Graph kann mit der Limit Order Liste zusammenarbeiten und zeigt Stop Loss / Take Profit sowie Call Levels. Die Eingaben in die Limit Order Liste müssen in den FX Graph übertragen werden. Eingaben in dem Graph (Ctrl – Mausklick und Rauf-/Runter bewegen der Maus) müssen in der Liste angezeigt werden
Erwartetes Resultat	Wird ein Call Level eingegeben, wird dieser mittels blauer Linie mit Verlauf angezeigt (Take Profit, Verlauf nach oben, Stop Loss Verlauf nach unten). Wird ein Stop Loss oder Take Profit eingegeben, so werden diese rot/grün dargestellt. Eingaben von Stop Loss / Take Profit mit der Maus müssen in der Liste angezeigt werden. Diese werden durch CTRL - Klick + Mausbewegung rauf-/runter gezeichnet. Eingabe des Call Levels mittels CTRL – Doppelklick (nur bei Typ Call Level möglich).
Resultat	Graph verhaltet sich wie oben beschrieben

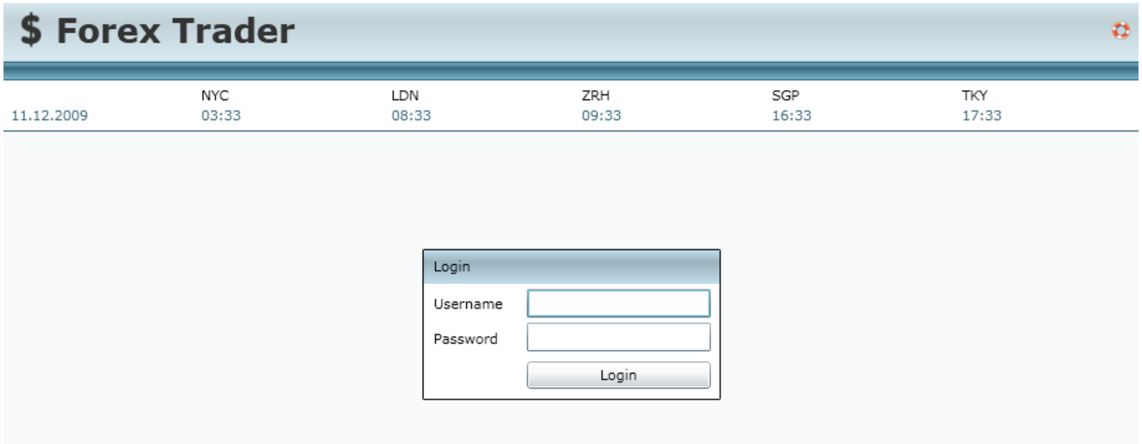
Testfall	**Limit Order speichern**
Beschreibung	Einen Limit Order soll man mittels klick auf den Send Limit Order Button speichern können. Die getätigten Eingaben werden dann gespeichert und würden vom Banksystem ausgewertet werden.
Erwartetes Resultat	Nach dem senden eines Limit Orders wird ein Fenster angezeigt, auf welchem man den Limit Order noch einmal bestätigen muss. Bestätigt man diesen so soll ein Datensatz in der Datenbank angelegt sein und dieser muss in der Limit Order Ticket-Liste erscheinen.
Resultat	Nicht Implementiert

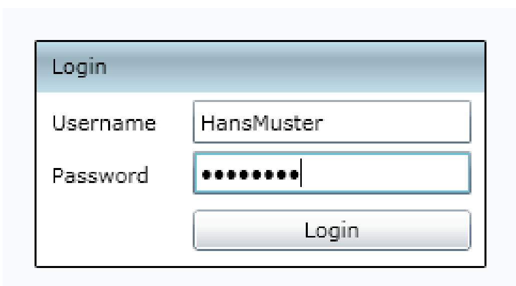
Foreign Exchange UI

Benutzerdokumentation

1 Anmeldung

Wenn sie die Webapplikation Forex Trader aufstarten, sehen sie folgenden Anmeldebildschirm, in welchem sie ihre Benutzerdaten eingeben und sich anmelden können.





Die Anmeldung am Forex Trader geschieht durch einen Benutzernamen sowie ein Passwort.
Die Eingabe bestätigen sie mit einem Klick auf Login oder durch Drücken der Enter Taste.

2 Übersicht

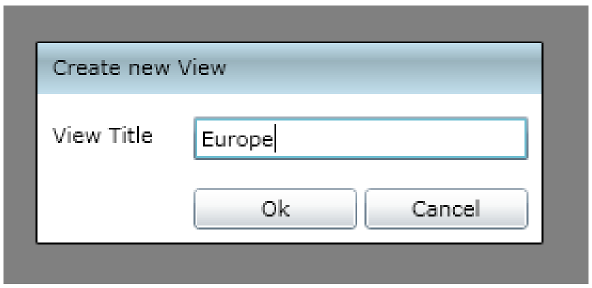

Uhrzeiten anderer Handelsplätze
SPOT Datum
Vom System abmelden
Hilfe anzeigen
$ Forex Trader
Logout
11.12.2009
NYC 03:15
LDN 08:15
ZRH 09:15
SGP 16:15
TKY 17:15
Spotdate 13.12.2009
Europe
EURUSD
Sell
Buy
1.49
23
02
1.49
26
1'000'000
USD
GBPCHF
Sell
Buy
1.66
60
04
1.66
65
1'000'000
CHF
EURCHF
Sell
Buy
1.53
40
03
1.53
44
1'000'000
CHF
Recent deals
Recent limited
Auswahl der Ebs Ansicht
Ebs Tile
Aktionen für Ebs Ansicht
- Neue Ebs Ansicht anlegen
- Ebs Ansicht speichern
- Ebs Ansicht löschen
- Der Ebs Ansicht ein neues Ebs Tile hinzufügen
Ansicht der letzten Deals / limitierten Aufträgen

3 Ansichten

3.1 Ansichtsauswahl

Die Auswahl der verschiedenen Ansichten geschieht über die Combobox ‚Auswahl der Ebs Ansicht‘. Nach dem Auswählen wird diese geladen und dessen Tiles in der Hauptansicht angezeigt





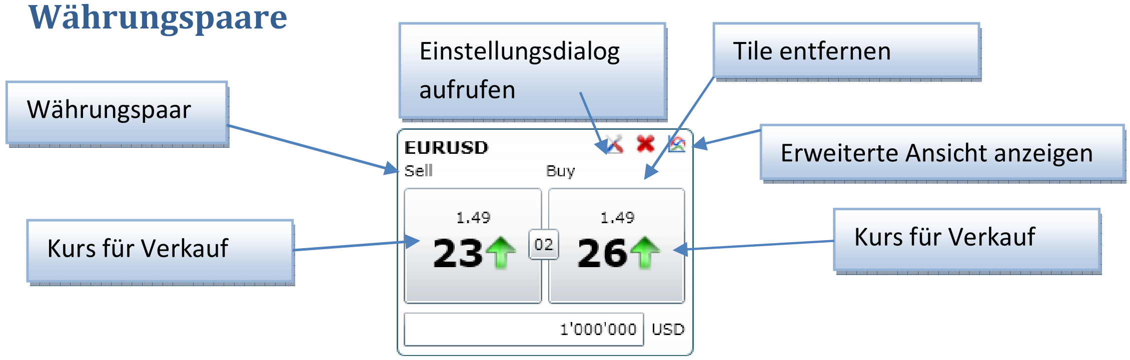
3.1.1 Neue Ansicht anlegen

Um eine neue Ebs Ansicht zu erstellen wählt man den ‚neue Ebs Ansicht anlegen‘. Nun öffnet sich ein Fenster, in welchem man den Namen der neuen Ansicht angeben kann. Bestätigt man diese Eingabe durch klicken des Ok Knopfes, so wird eine neue Ansicht, falls noch keine unter diesem Namen abgespeichert wurde, gespeichert. Wurde bereits eine Ansicht mit demselben Namen angelegt, kann diese nicht gespeichert werden und es wird eine Fehlermeldung angezeigt.

3.1.2 Ansicht entfernen

Das Entfernen einer Ansicht erreicht man, wenn man die gewünschte Ansicht aus der Ansichtsauswahl auswählt und, wenn diese geladen ist, den ‚Ebs Ansicht löschen‘ Knopf betätigt.

4 Währungspaare

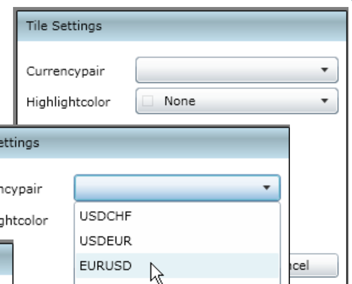


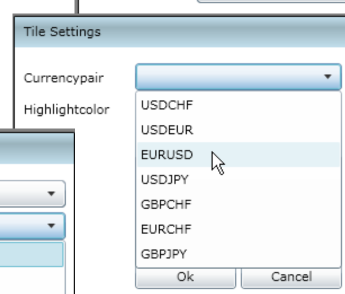
4.1 Verwalten

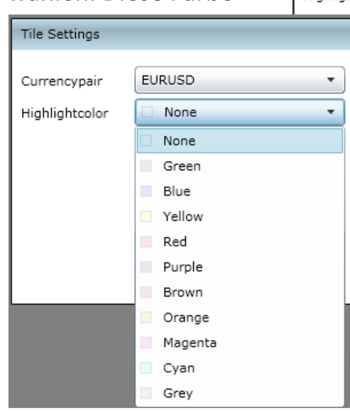
4.1.1 Hinzufügen



Um in der Ansicht ein neues Währungspaar hinzuzufügen, klickt man auf den ‚Der Ebs Ansicht neues Ebs Tile hinzufügen' Knopf, welcher die nebenstehende Ansicht anzeigt.
In dieser Ansicht wählt man das gewünschte Währungspaar aus und man kann, wenn man möchte, dem Währungspaar eine Farbe, welche das Tile hervorhebt, wählen. Diese Farbe wird dezent über das Tile gezeichnet, um wichtige Währungspaare besser im Auge zu behalten.







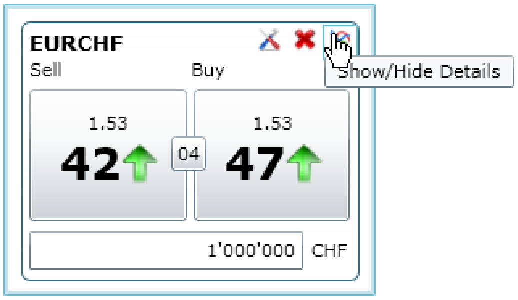
4.1.2 Entfernen

Das entfernen eines Tiles geschieht durch einen Klick auf den ‚Tile entfernen' Knopf.

4.1.3 Einstellungen

Der Einstellungsdialog bietet dieselben Möglichkeiten wie beim hinzufügen eines Währungspaares (siehe Hinzufügen)

4.2 Details



Um die Details eines Währungspaares anzusehen, klickt man auf Show/Hide Details.

Die Detailansicht enthält erweiterte Erfassungsmöglichkeiten für Spot/Outright sowie Swap-Geschäfte sowie einen Graphen.





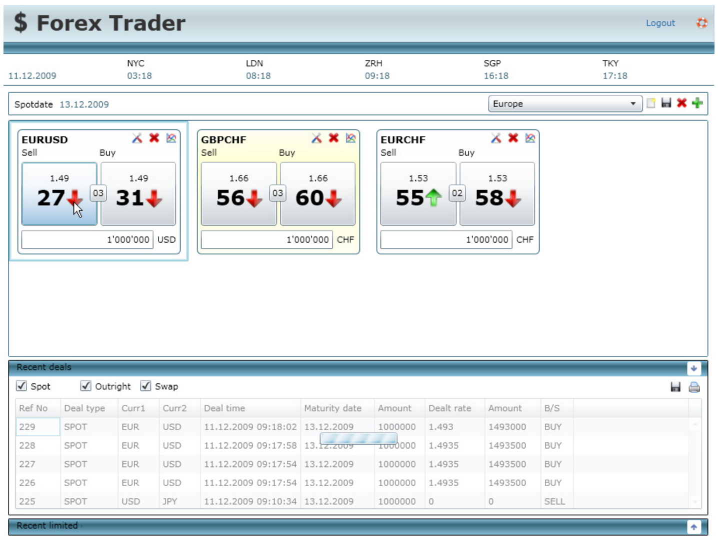
Die Skalierung des Graphen nimmt man mit dem Slider oder durch das Scrollrad der Maus vor. Es reicht von einer detaillierten Darstellung, bis hin zu einer Jahresübersicht

5 Handeln

5.1 SPOT/Forward

Das Handel von Währungen geschieht über den Verkaufen- sowie den Kaufen-Knopf. Der Handel wird mit dem aktuellen Kurs der Währung, sowie dem eingegebenen Betrag vollzogen und erscheint in der Liste der Deals.





Wird ein Handel getätigt, wird die Recent deals-Liste aktualisiert

6 Limitierte Aufträge

Um die limitierten Aufträge zu erfassen, wählt man die ‚Limit Order View' in der Detailanzeige eines Tiles. Diese öffnet dann das Erfassen der limitierten Aufträge.





Diese Ansicht zeigt im Fussbereich des Graphen eine Eingabemaske. In dieser kann man nun einen solchen Auftrag erfassen. Die Erfassung kann aber auch grafisch erfolgen.

6.1 Grafisches Erfassen von limitierten Aufträgen

Die Erfassung von einem limitierten Auftrag – OCO. Zuerst wird die Take Profit (Tp) Rate gesetzt. Dies geschieht indem man in dem Graph an die gewünschte Position fährt. Hat man die gewünschte Position erreicht, so drückt man nun die CTRL (oder auch STRG) und Klickt dabei mit der linken Maustaste, hält diese gedrückt und zieht die Maus nach oben.



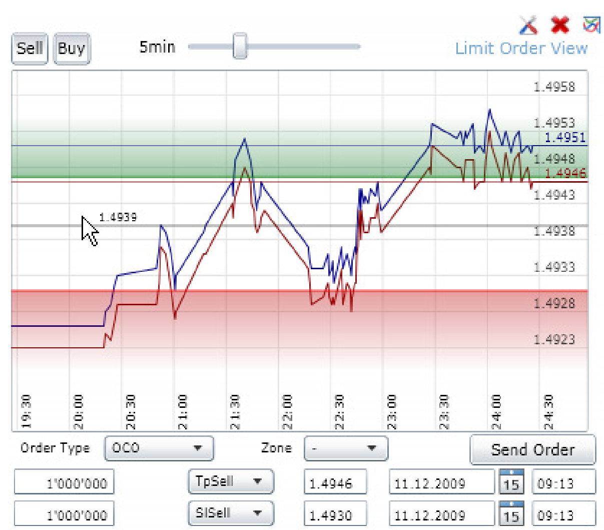


Erscheint ein grüner, vertikaler Balken hat man alles richtig gemacht. Lässt man nun die Maustaste los wird der Startwert der Linie als Rate für den Take Profit eingetragen und ein Verlauf wird im Graph dargestellt



Mit dem gleichen vorgehen kann man nun auch einen Stop Loss Wert hinzufügen. Dies geschieht wiederum durch drücken der CTRL Taste und Klicken der linken Maustaste, diesmal aber durch ziehen der Maus nach unten.

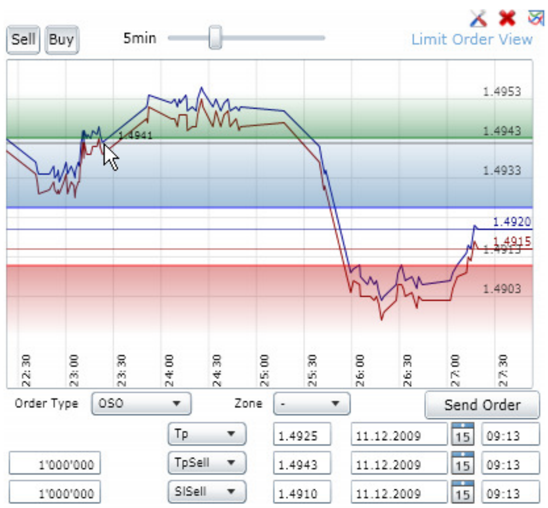
Der Fertig erfasste Auftrag wird nun dargestellt. Man kann die Werte der Raten bei Bedarf ändern.
Klickt man nun den Send Order Knopf, so wird der limitierte Auftrag abgesendet und die Eingabe geleert.



6.1.1 Erfassen von Call Level

Die Call Level können nur in den beiden limitierten Auftragstypen OSO (One starts the other) sowie Call-Level vorkommen.

Die Call Level Erfassung wird mit derselben Methode getätigt, wie bereits das Erfassen von StopLoss/TakeProfit. Einziger Unterschied, die Linien sind blau und die Maus wird nicht nach oben bzw. unten gezogen, sondern nach links bzw. rechts.

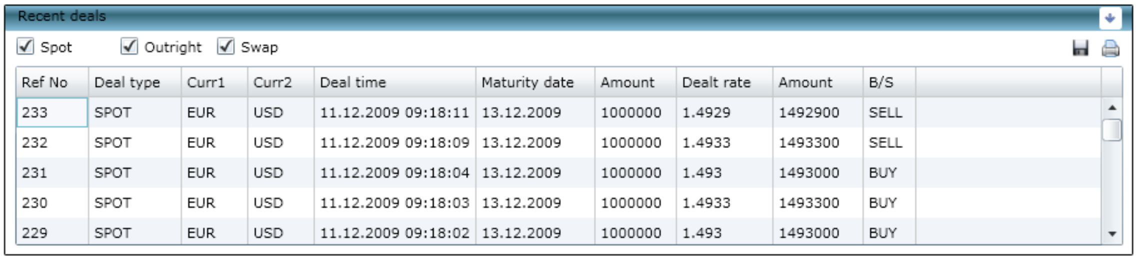




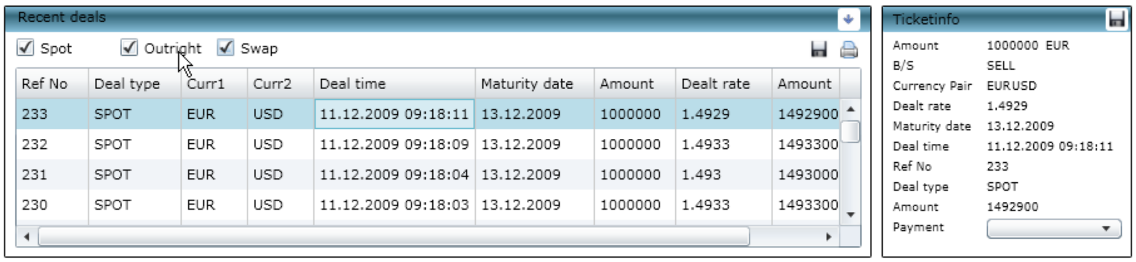
Dieses veränderte Verhalten ist nötig, um beim Erfassen keinen Konflikt zwischen Take Profit / Stop Loss und Call Level zu haben.

7 Recent Deals

In der Liste der Recent Deals werden die letzten 100 getätigten Deals aufgelistet. Wählt man einen Eintrag aus, so wird auf der rechten Seite eine Detailansicht angezeigt, in welcher man den Zahlungsweg ändern könnte.



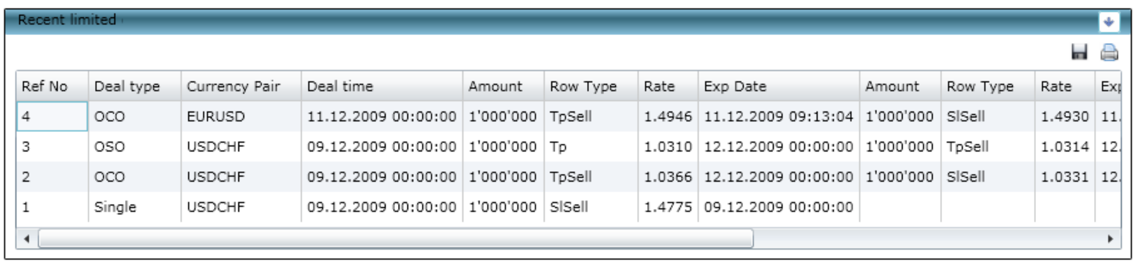
Ref No	Deal type	Curr1	Curr2	Deal time	Maturity date	Amount	Dealt rate	Amount	B/S
233	SPOT	EUR	USD	11.12.2009 09:18:11	13.12.2009	1000000	1.4929	1492900	SELL
232	SPOT	EUR	USD	11.12.2009 09:18:09	13.12.2009	1000000	1.4933	1493300	SELL
231	SPOT	EUR	USD	11.12.2009 09:18:04	13.12.2009	1000000	1.493	1493000	BUY
230	SPOT	EUR	USD	11.12.2009 09:18:03	13.12.2009	1000000	1.4933	1493300	BUY
229	SPOT	EUR	USD	11.12.2009 09:18:02	13.12.2009	1000000	1.493	1493000	BUY



Ref No	Deal type	Curr1	Curr2	Deal time	Maturity date	Amount	Dealt rate	Amount
233	SPOT	EUR	USD	11.12.2009 09:18:11	13.12.2009	1000000	1.4929	1492900
232	SPOT	EUR	USD	11.12.2009 09:18:09	13.12.2009	1000000	1.4933	1493300
231	SPOT	EUR	USD	11.12.2009 09:18:04	13.12.2009	1000000	1.493	1493000
230	SPOT	EUR	USD	11.12.2009 09:18:03	13.12.2009	1000000	1.4933	1493300

8 Recent Limited Orders

Die Anzeige der limitierten Aufträge zeigt ebenfalls die letzten 100 erfassten Aufträge.



Ref No	Deal type	Currency Pair	Deal time	Amount	Row Type	Rate	Exp Date	Amount	Row Type	Rate	Ex
4	OCO	EURUSD	11.12.2009 00:00:00	1'000'000	TpSell	1.4946	11.12.2009 09:13:04	1'000'000	SlSell	1.4930	11.
3	OSO	USDCHF	09.12.2009 00:00:00	1'000'000	Tp	1.0310	12.12.2009 00:00:00	1'000'000	TpSell	1.0314	12.
2	OCO	USDCHF	09.12.2009 00:00:00	1'000'000	TpSell	1.0366	12.12.2009 00:00:00	1'000'000	SlSell	1.0331	12.
1	Single	USDCHF	09.12.2009 00:00:00	1'000'000	SlSell	1.4775	09.12.2009 00:00:00				

Foreign Exchange UI

Summary

1 Einleitung

Es gibt auf der Welt keinen zentralen Platz für Währungshandel. Dieser wird in den Hauptzentren auf der ganzen Welt (London, New York, Tokyo, Zürich, Frankfurt, Hong Kong, Singapur, Paris, Sydney) verwaltet. Der Devisenhandel ist 24 Stunden pro Tag und 5 Tage die Woche offen. Es ist der grösste Finanzmarkt der Welt und jede Firma bzw. jedermann kann in diesem Markt handeln. Gehandelt wird mit unterschiedlichsten Applikation wie z.B. dem Meta Trader. Ein mächtiges Tool, welches wohl vor allem für erfahrene Profis gemacht ist.
Am häufigsten gehandelt werden die Währungen der grössten Wirtschaftsräume (USD, EUR, YEN) sowie Währungen aus Ländern mit starken Finanzplätzen (z.B. GBP, CHF).
Die Credit Suisse betreibt eine Online Applikation (basierend auf Java Applets) für den Devisenhandel. Es ist möglich die aktuellen Kurse live anzuschauen und sofort Trades zu erfassen oder auch Limited Orders zu platzieren.

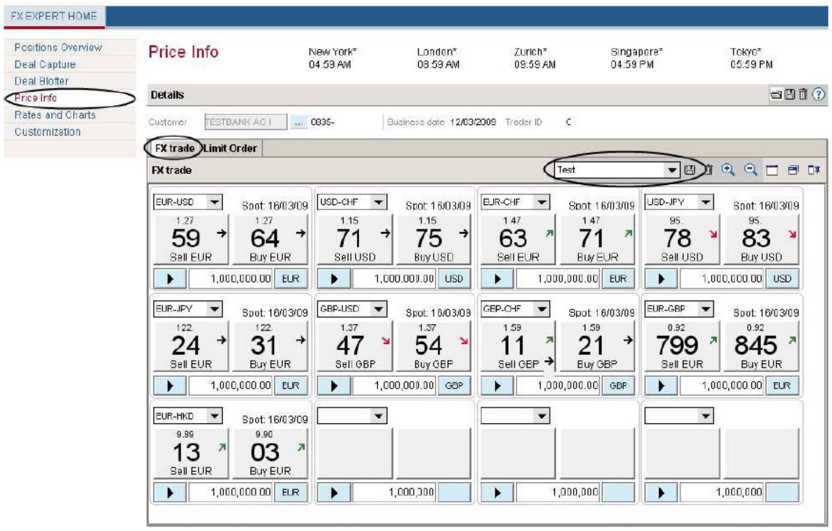
1.1 Projektbeschreibung

Ziel ist es, die Möglichkeiten von Silverlight anhand einer Applikation für den Devisenhandel zu erforschen. Die Applikation sollte modular aufgebaut sein, geringe Latenz garantieren, kurze Entwicklungszeiten für Folgearbeiten haben und die Wartung sollte auch möglichst einfach sein.
Das User Interface soll verbessert werden. um intuitives Arbeiten zu ermöglichen. Es sollen mehrere aktuelle Kurse angezeigt werden können, die für den Benutzer personalisierbar sind. Ausserdem sollen die wichtigen Stellen (Pips) hervorgehoben und spezielle Kurse besonders gekennzeichnet werden können.
Weiter sollen normale Geschäftsaufträge sowie limitierte Aufträge wie z.B. OSO und OCO erfasst werden können. Das Erfassen dieser limitierten Aufträge sollte möglichst einfach sein.
Auf die Sicherheit soll primär nicht geachtet werden.

1.2 Bestehende Lösungen

1.2.1 Aktuelle Lösung

https://www.credit-suisse.com/



1: Ansicht der abonnierten Kurse in FX Expert

Die aktuelle Lösung der Credit Suisse besteht aus Java Applets und nennt sich FX Expert. Der Aufbau des User Interfaces kann zu grossen Teilen davon abgeleitet und verbessert werden.

1.2.2 Morgan Stanley Matrix

http://www.morganstanley.com/matrixinfo/

Die Lösung von Morgan Stanley ist mit Adobe Flash gelöst und besticht vor allem durch sein schönes Design. Weitere Features dieser Losung sind aktuelle Videos von Experten mit Markteinschätzungen und Empfehlungen.



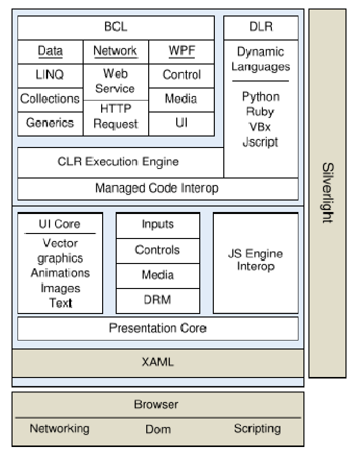
2: Ansicht der Kurse in der Lösung von Morgan Stanley

2 Silverlight

Microsoft Silverlight ist ein Plug-In für Webbrowser, welches auf .NET basiert, die Ausführung von Rich Internet Applications ermöglicht und damit über die Möglichkeiten des klassischen HTML hinausgeht. Dem Anwender werden z.B. Drag and Drop, 3D-Effekte und Animationen zur Verfügung gestellt. Zudem erfolgt meist eine schnellere Reaktion als bei klassischen HTML-basierten Anwendungen, da die Anwendung auf dem lokalen Rechner ausgeführt wird und somit nicht auf die Reaktion eines Servers gewartet werden muss. Silverlight ist als proprietäres, programmierbares Plug-in für Windows und Apple Macintosh verfügbar. Für Linux wird von Novell mit Zustimmung und Unterstützung von Microsoft Moonlight angeboten. Bei unserer Arbeit haben wir mit Silverlight 3 gearbeitet. Im November 09 ist die erste Beta zu Silverlight 4 aufgetaucht.

2.1 Aufbau

Silverlight-Anwendungen werden vom Webserver heruntergeladen und auf dem Client typischerweise im Browser ausgeführt (kann auch „Out-of-Browser“ betrieben werden). Die Kommunikation der Anwendung mit dem Webserver erfolgt mittels HTTP-GET, REST oder Webservices. Für die Programmierung mittels .NET eignen sich unter anderem die ADO.NET Data Services, die Datenbanken automatisch als Webdienst für einen Silverlight-basierten RIA-Client bereitstellen können.
Silverlight ist hinsichtlich seiner UI-Layer abgeleitet aus der Windows Presentation Foundation. Hauptbestandteil der vektorbasierten Grafikdarstellung und der Gestaltung von Anwendungsoberflächen ist das universelle und textbasierte XML-Format XAML.



3: Aufbau von Silverlight

2.2 Erfahrungen

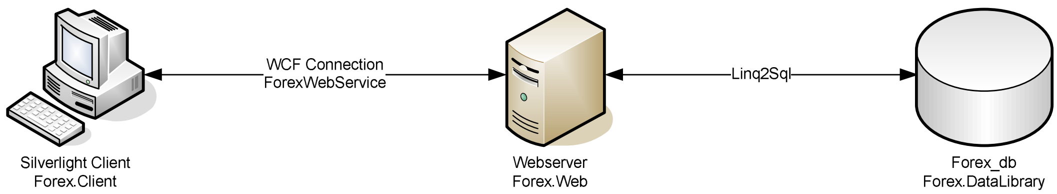
Während unserer Semesterarbeit sind uns Dinge aufgefallen, die von Silverlight (noch) nicht unterstützt werden.

- Animieren des Margin-Properties
- Fehlende Unterstützung des MVVM Patterns (konnte nur mit Hilfe von Zusatzlibraries ermöglicht werden)
- WCF Service funktioniert nur mit „basicHttpBinding“ (Kein Duplex, Keine Security, keine Transactions)

Silverlight 4 bietet einige Verbesserung. Aktuell ist die erste Beta verfügbar. Laut Angaben von Microsoft soll die Unterstützung von WCF stark verbessert werden, was darauf hoffen lässt, dass auch andere Bindings unterstützt werden. Weiter besteht die Möglichkeit, dass Silverlight 4 Commands standardmässig unterstützt.

3 Ergebnis

3.1 Architektur



4: Übersicht des Kommunikationsablauf zwischen Client, Server und Datenbank

3.2 Testbed

Das Testbed beinhaltet den WCF-Service, welcher für die Kommunikation mit dem Client zuständig ist und die Rates generiert sowie die Website, welche der Silverlight Client zum Download anbietet.

3.2.1 ForexWebService

Der Service stellt die Rates zur Verfügung, welche vom Client abonniert werden können. Da Silverlight nur das „basicHttpBinding“ unterstützt, muss der Client jedes Mal seine abonnierten Rates abfragen und kann diese nicht in einem Stream vom Server erhalten.
Weiter handelt der Service die Benutzerverwaltung (Login, Logout, Views, etc.) und platziert die Trades.

3.2.2 Andere Techniken

Eine andere Möglichkeit wäre es über den ClientStack zu gehen und vom Server aus ständig Messages zu schicken. Dies wäre wohl schneller, aber auch hier müsste man immer einen Request schicken, was man genau will. Zudem wäre die Sicherheit nicht so gut zu handeln, wie bei einem WCF-Service.

3.3 User Interface

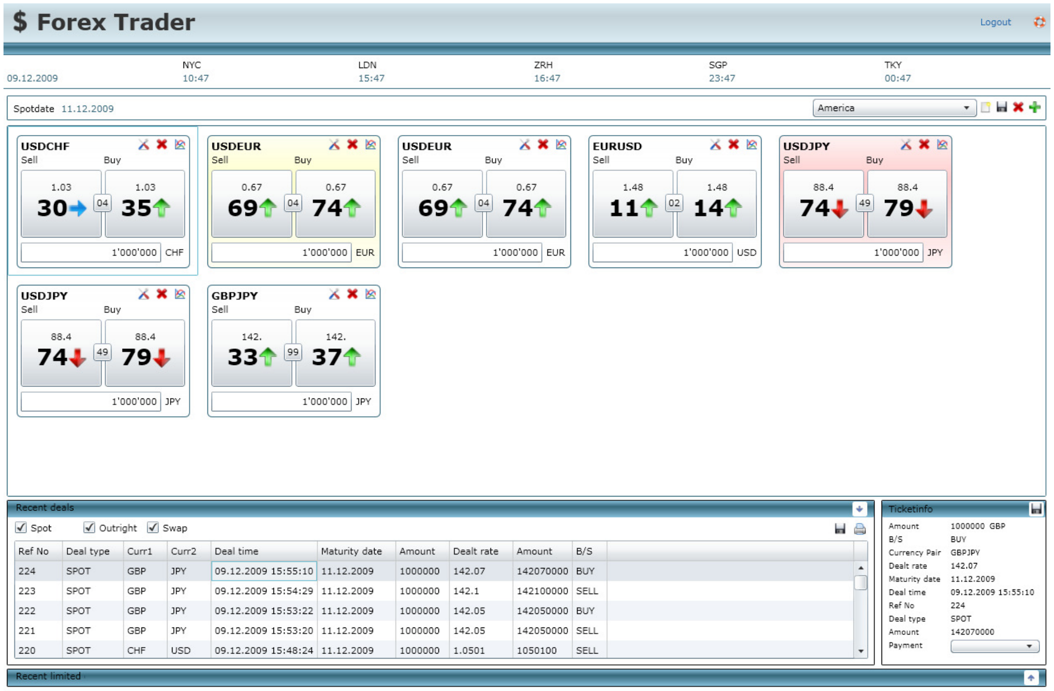
3.3.1 Aufbau

Das User Interface ist grob nach dem MVVM (Model-View-ViewModel) Pattern erstellt. Durch die fehlende Unterstützung seitens Silverlight, wurde auf eine Zusatzlibrary (SLExtensions) zurückgegriffen, welche vor allem wegen den Commands gebraucht wurde.

3.3.2 Design

Das externe Design wurde zu grossen Teilen aus der Java Applet Version übernommen, jedoch mit ein paar Ergänzungen und Veränderungen wie beispielsweise das Erfassen von Aufträgen in einem Graph oder nur eine Ansicht für normale und limitierte Aufträge. Zudem wurde das User Interface optisch auf einen moderneren Stand gebracht. Dies wurde erreicht durch schlichteres Design, welches wichtige Controls hervorhebt (z.B. durch Abrundungen oder Farbverläufe) und der Vereinheitlichung durch vordefinierte Styles, welche einfach angepasst werden können. Auch alle statischen Ressourcen (Texte, Bilder) sind vordefiniert und können schnell und einfach geändert werden.

3.3.2.1 Hauptansicht



5: Hauptansicht des Forex Traders mit allen abonnierten Rates und den letzen getätigten Trades

In der Hauptansicht sieht man alle aktuell abonnierten Rates auf sogenannten Tiles. In dieser Ansicht werden auch die letzten 100 Deals angezeigt sowie die persönlichen Views verwaltet.

3.3.2.2 Tiles



6: Ansicht des aufgeklappten Tiles mit Live-Graph und Live-Rate

Auf einem Tile wird der aktuelle Kurs des gewünschten Currencypairs angezeigt. Pro Tile gibt es 3 Ansichten. In der Standardansicht wird nur der Kurs angezeigt und es können direkt SPOT-Trades ausgeführt werden. In der ersten aufgeklappten Ansicht sieht man den Live Graph und es besteht die Möglichkeit SWAPs und Outrights zu platzieren. In der dritten Ansicht kann man Limited Orders erfassen. Die Limited Orders können direkt im Graph durch klicken und ziehen markiert werden.

3.4 Fehlendes

3.4.1 Durch die fehlende Zeit konnten wir ein paar Pendenzen nicht abarbeiten.Testbed

Folgende Funktionen fehlen im Testbed bzw. beim Service

- Speichern von Ticketdetails
  Der geänderte Zahlungsweg kann nicht gespeichert werden.
- Speichern von SWAPs und Outrights

3.4.2 User Interface

Folgende Funktionalitäten fehlen im User Interface

- Generierung eines PDFs der letzten Deals / Limited Orders
  Es kann zwar ein PDF heruntergeladen werden, jedoch zeigt dieses nur an, dass das Herunterladen funktioniert. Es wird kein benutzerspezifisches PDF generiert. Dies könnte man einfach mit einer zusätzlichen Library generieren.
- Festlegen mit welchem Konto man handelt
  Im Moment gibt es pro Benutzer ein Konto
- Bearbeiten von Limit Orders
  Gespeicherte Limited Orders können nicht mehr verändert/gelöscht werden.

4 Erkenntnisse

4.1 Server

Der Server war von der Logik her nicht das Hauptproblem. Schwieriger war eher die Entscheidung, will man nun mit zwei Services arbeiten (SOAP und WCF), was zu Komplikationen mit der Session führen könnte oder nur mit einem Service (WCF), wo man zwar das Session Problem nicht hat, jedoch auch keine eine Trennung von den generierten Rates und der ganzen Benutzersteuerung. Wir entschieden uns für nur einen Service, da dies vom Sicherheitsaspekt wohl mehr Sinn ergibt, weil man sonst mit einer Art Proxy arbeiten müsste.

4.2 Silverlight

Da Silverlight eine, der am schnellsten wachsenden Internettechnologieplattformen, ist, war die Arbeit mit Silverlight höchst interessant und lehrreich. Man merkte auch, dass es Silverlight noch nicht allzu lange gibt und ständig erweitert wird. Während unserer Entwicklungsphase starteten wir mit dem Toolkit vom Juli, stellten dann um auf Oktober und später kam dann noch das Toolkit vom November und die Silverlight 4 Beta, welche wir jedoch nicht eingesetzt haben. Die Verbesserungen vom Juli zum Oktober waren eigentlich minimal. Einzig die Charting API wurde verbessert. An unserem bestehenden Code mussten wir nichts ändern.
Am Neusten war für uns die ganze Beschreibung des User Interfaces mit XAML, da wir das noch nie gemacht hatten und die Module, welche dies behandeln (MsTech, IntT) noch vor uns liegen. Zum Glück ist die Silverlight/WPF Community schon sehr gross, da Microsoft Silverlight extrem pusht und die Anzahl Arbeitsplätze in diesem Bereich stark zunimmt.

4.2.1 Performance

Die Performance von Silverlight ist relativ gering. Anfangs hatten wir noch ohne Performanceoptimierung gearbeitet und es kam schnell vor, dass die Prozessorauslastung über 30% betrug. Nachdem wir das Zeichnen von der CPU auf die GPU ausgelagert hatten, sank dieser Wert drastisch in den einstelligen Prozentbereich. Problematisch ist es nur dann, wenn mehrere Graphen gleichzeitig angezeigt werden. Bei diesen Graphen sind die Performanceprobleme aber auf die Animationen abzuschieben. Die Animationen benötigen sehr viele Ressourcen, vor allem dann, wenn die Framerate der Silverlight-Applikation nicht gesetzt ist, da Silverlight standardmässig 64 Frames pro Sekunde zeichnet, was zu viel des Guten ist.

4.2.2 Security

Silverlight 3 bietet wohl für eine Bank noch zu wenige Sicherheiten, doch in Silverlight 4 wird die Sicherheit markant erhöht. Wie schon beschrieben, wird die Unterstützung von WCF verbessert, was wohl auch zu erhöhter Sicherheit führen wird.
Ansonsten kann man auch die ganze Applikation (Client, Service, Website) in einer SSL-Session betreiben.

4.3 Bankengeschäft

Da wir keine Banker sind, war es für uns nicht leicht diese Geschäftsprozesse richtig zu verstehen. Um diese Prozesse richtig nachvollziehen zu können, muss man im Bankenumfeld tätig sein. Vor allem die limitierten Aufträge wie OCO und OSO bereiteten uns grosses Kopfzerbrechen. Dank einem Buch, welches wir organisierten, war der Einstieg deutlich einfacher.

4.4 Weiteres

Durch den Besuch der Moduls Microsoft Technologien und des für dieses Modul obligatorische Mini-Projekt, sah man noch viele Verbesserungsmöglichkeiten für unsere Applikation, welche jedoch zu diesem Zeitpunkt teilweise nicht mehr berücksichtigt werden konnten.

- Linq to Sql (Implementiert)
  Der Zugriff auf die Datenbank erfolgt über Linq to Sql. Die Datenbank wird gemappt, sodass über LINQ (Language Integrated Query) auf die Datenbank zugegriffen werden kann. Dies erleichtert die Lesbarkeit des Codes und Änderungen können einfacher implementiert werden.
- DTOs (Implementiert)
  Über die WCF Schnittstelle werden nicht direkt „Linq to Sql"-Objekte übertragen, sondern sogenannte (DTOs) DataTransferObjects. Dies führt dazu, dass nur die für den Client wichtigen Daten übertragen werden.
- WCF Interface (Nicht Implementiert)
  Am besten wäre es, wenn der OperationContract für den WCF Service als Library für alle Projekte verfügbar wäre, damit alle Projekte immer den gleichen Stand haben. Da Silverlight jedoch nur andere Silverlight Projekte als Referenzen akzeptiert, haben wir darauf verzichtet.

Foreign Exchange UI

Glossar

1 Dokumentinformationen

1.1 Änderungsgeschichte

Datum	*Version*	*Änderung*	*Autor*
17.09.09	0.1	Dokument erstellt	dh
18.09.09	0.2	Finanzbegriffe	dh
18.09.09	0.3	Technologiebegriffe	ds
13.10.09	0.4	Bindings	dh
06.11.09	0.5	Forex Trader	dh
11.12.09	1.0	Review / Version 1	dh

1.2 Referenzen

Nr.	*Dokument*	*Autor*
1	Orders, Dokument Credit-Suisse https://entry.credit-suisse.ch/csfs/p/cb/de/dev_zinsen/media/pdf/dzs_stoploss_de.pdf	CS
2	Brief- und Geldkurs Einträge in Wikipedia.org http://de.wikipedia.org/wiki/Briefkurs http://de.wikipedia.org/wiki/Geldkurs	WIKI
3	Termingeschäft, Dokument Credit-Suisse https://entry.credit-suisse.ch/csfs/p/cb/de/dev_zinsen/media/pdf/dzs_termingeschaeft_de.pdf	CS

2 Forex Trader Begriffe

Begriff	*Erklärung*
Tile	Ein Tile ist eine Ansicht eines Währungspaares und zeigt die buy sowie die sell Rate. Zudem enthält ein Tile auch weiter Elemente (DetailTile)
DetailTile	Das DetailTile ist die Detailansicht eines Tiles. Es bietet zu der normalen Ansicht noch erweiterte Deal-Einstellungen sowie einen Verlaufsgraphen.
Deal	Einen Handel zwischen zwei Währungen, wobei der Typ des Handels SPOT, SWAP sowie Outright sein kann.

3 Finanzbegriffe

3.1 Allgemein

Begriff	*Erklärung*
Big Figures	Die Big Figures sind die ersten drei Ziffern eines Kurses. Sie sind für den Devisenhändler meist uninteressant, da diese in der Regel bekannt sind und sich nicht so schnell ändern. Bsp. EURCHF **1.51**661
Pips	Der Kurs zwischen zwei Währungen besitzt zwei sehr wichtige Zahlen, die sogenannten Pips. Sie befinden sich an Position 4 und 5 der Zahl, mit diesen wird ‚Geld gemacht'. Bsp. EURCHF 1.51**66**1, EURJPY 133.**80**1
Request for Quote (RFQ)	Anfrage eines Kurses zwischen zwei Währungen (z.B. EURCHF)
FOREX Quotes	Kurs eines Währungspaars

3.2 Orders

Begriff	*Erklärung*
Briefkurs	Kurs, zu dem ein Marktteilnehmer bereit ist, seine Devisen zu verkaufen. [2]
Call Level	Ein Auftrag, welcher nur eine Orientierung des Kunden zur Folge hat. Bsp.:Der Kunde möchte bei einem CHFUSD Kurs von 1.10 orientiert werden. Steigt nun der Kurs auf diesen Wert, wird dieser benachrichtig ohne etwas zu kaufen oder verkaufen.
Geldkurs	Kurs, zu dem ein Marktteilnehmer bereit ist, Devisen zu kaufen. [2]
One cancels the Other (OCO)	Ähnlich OSO. Umfasst zwei Aufträge; beim Ausführen des einen wird der andere beendet. Damit kann von einer vorteilhaften Kursbewegung mit gleichzeitiger Absicherung profitiert werden.
One starts the Other (OSO)	Kombination zweier Aufträge in Form von 'sell after buy' oder 'buy after sell'. Ein Auftrag wird überwacht. Sobald dieser ausgeführt wurde, wird ein zweiter gestartet (meist um 1. Geschäft gewinnbringend einzudecken). [1]
Stop Loss Aufträge	Schützen bestehende Positionen vor negativen Auswirkungen von Kursbewegungen. Ähnlich Take Profit sell. [1]
Take Profit (buy/sell)	Stellt sicher, dass der Handel bei einer gewissen Marke abgeschlossen oder ein neuer Handel gemacht wird. Bsp.: Kauf von CHFUSD zu einem Kurs von 1.10, nach kurzer Zeit steigt der Kurs auf 1.16, fällt danach aber auf 1.05. Take Profit wäre ein Trigger, der zum Beispiel bei einem fallenden Kurs auf 1.15 verkauft.

3.3 Termingeschäft

Begriff	*Erklärung*
Termingeschäft (Forward/Outright)	Jede konvertible Währung kann auf jeden gewünschten Termin gekauft oder verkauft werden. Der Terminkurs besteht aus dem Kassakurs und dem Swapsatz.[3]
Kassageschäft (Spot)	Ein Termingeschäft, welches in der Regel eine zweitägige Laufzeit besitzt; heute den Handel tätigen, in zwei Tagen bezahlen.

3.4 Swapgeschäft

Begriff	*Erklärung*
Swap	Ein Swap ist ein Finanzmarktgeschäft, das aus einem Devisen-Kassageschäft (Spot) und einem Devisentermingeschäft (Forward) besteht. Dabei werden zwei Währungen per Spot gegeneinander getauscht und zu einem späteren Zeitpunkt wieder zurückgetauscht. Beide Transaktionen eines Devisenswaps werden gleichzeitig und mit derselben Gegenpartei abgeschlossen.

4 Technologiebegriffe

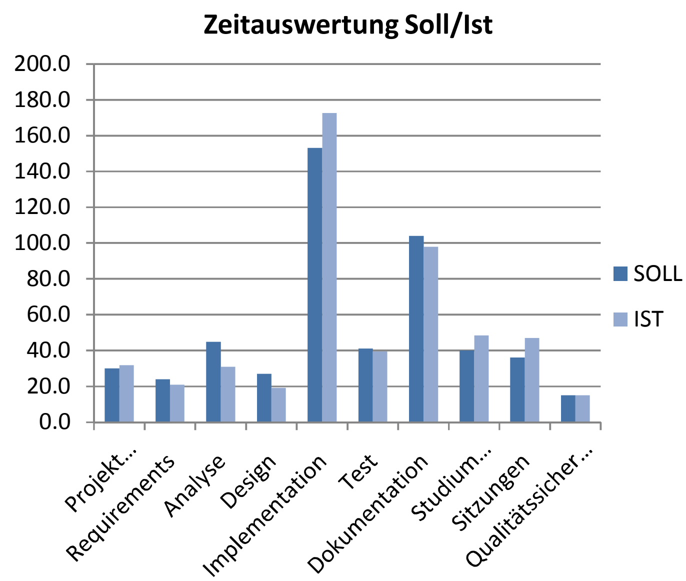
4.1 Allgemein

Begriff	*Erklärung*
RIA	Rich Internet Application; beschreibt Anwendungen, welche Internet-Techniken und eigenständige, intuitive Benutzeroberflächen nutzt.
Silverlight	Microsoft Silverlight ist eine Erweiterung für Webbrowser, welche die Ausführung von Rich Internet Applications (RIAs) ermöglicht und damit über die Möglichkeiten des klassischen HTML hinausgeht (z.B. Drag & Drop, Animationen, etc.)
WCF	Windows Communictaion Foundation; Die Windows Communication Foundation soll in Microsoft Windows eine neue, dienstorientierte Kommunikationsplattform für verteilte Anwendungen werden. Microsoft will hier viele Netzwerk-Funktionen zusammenführen und den Programmierern solcher Anwendungen standardisiert zur Verfügung stellen.
WPF	Windows Presentation Foundation; WPF stellt ein umfangreiches Modell für den Programmierer bereit. Dabei werden die Präsentation und die Geschäftslogik getrennt, dies wird vor allem durch die Auszeichnungssprache XAML unterstützt. Silverlight ist die Plattform, welche explizit für das Web entwickelt wurde.
XAML	XAML beschreibt eine Oberflächen-Hierarchien deklarativ als XML-Code und wird für Silverlight sowie für WPF verwendet.
Binding	Anbindung von Daten an einen Control / an einen Service. Silverlight: Binding bindet Daten aus dem Datenkontext an ein Control. WCF: Binding gibt an, auf welche Art der Austausch zwischen WCF Service und Client abgehandelt werden soll. Bsp.: BasicHttpBinding – Informationsaustausch über das http Protokoll.

4.2 Entwicklungswerkzeuge

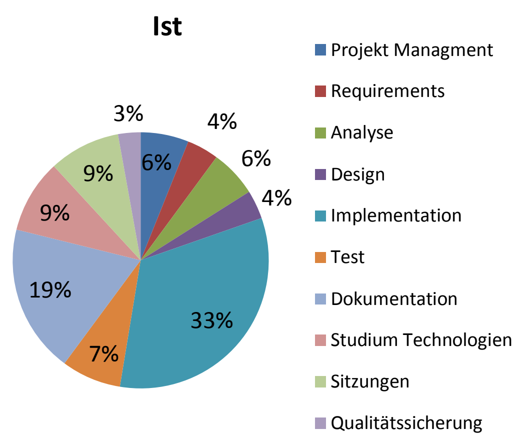
Begriff	*Erklärung*
Expression Blend	Entwicklungsumgebung für plattformübergreifende Webanwendungen (RIAs) auf Basis von Silverlight und XAML.

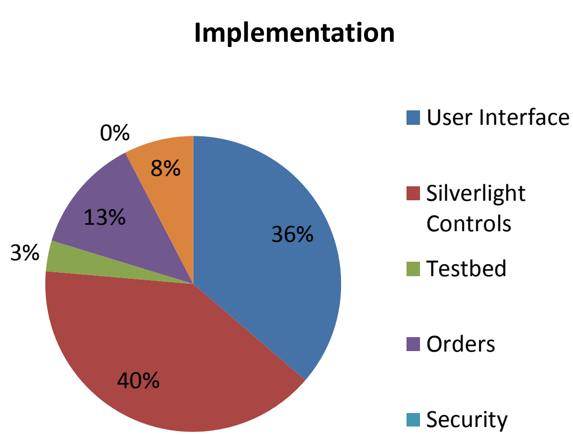
Zeitauswertung



Die Übersicht über die Soll/Ist Zeitauswertung zeigt ein relativ ausgeglichenes Ergebnis. Die Implementation haben wir zwar grosszügig geplant, trotzdem haben wir dies noch ein wenig unterschätzt.

Darstellung des Ist-Zustandes. Mit einem Drittel hat die Implementation, wie bereits oben ersichtlich, am meisten Zeit benötigt. Zusammen mit der Dokumentation machen diese Punkte über die Hälfte unserer Arbeit aus. Für die Qualitätssicherung haben wir mit nur 3% am wenigsten Zeit aufgewendet.





Die Auswertung des Ist-Zustandes der Implementation zeigt, dass vor allem das User Interface sowie das Implementieren der Silverlight Controls viel Zeit in Anspruch nahm, dies ist aber logisch.
Mit 8% haben wir den Code auch relativ oft durchgeschaut.
Die Security schlägt mit 0% zu Buche, da wir dafür keine Zeit mehr hatten.

Sitzungsprotokolle

1 Sitzungsprotokoll vom 15.09.09

1.1 Teilnehmer

- Prof. Dr. Markus Stolze
- Dr. Beat Liver
- Daniel Häfliger
- Dominik Süsstrunk

1.2 Traktanden

- Kickoff Meeting

1.3 Beschlüsse

- Technologie: Silverlight 3.0
  - Security – Zeigen was sicherheitstechnisch möglich ist
- Kennenlernen FOREX Begriffe
  - Swap, Spot, limited order, order, forward
- Überwachungsaufträge
  - System Überwacht Call Level (Trigger) -> SMS/E-Mail Benachrichtigung
  - OsO / OcO
  - take profit/profit loss
  - Request for quote (Kursanfrage)
- Bisheriges Handling
  - Click -> Done
  - Zeitscheibe (z.B. 5s) für Privatanwender
  - EBS – Tableau
- Gesuchte Lösung
  - Kurse in Grid anzeigen
    - Ev. zoombar
    - Einzelne Fenster vergrössern & Zusatzinformationen (z.B. Chart) anzeigen
    - Werden gestreamt, Anmeldung an streams für jede Währung (simulieren)
  - OsO/OcO Erfassung
    - ‚IQ Test' bestehen „Wieviele Anläufe bis richtig erfasst"
    - Ev. In Chart Call Levels setzen (Usability testen)
- Vorgehen
  - Zeitplan
  - Einarbeiten
    - Foreign Exchange
    - Bestehende Lösung
    - Silverlight
  - Dokumentstruktur / Projektverwaltung einrichten

2 Sitzungsprotokoll vom 23.09.09

2.1 Teilnehmer

- Prof. Dr. Markus Stolze
- Daniel Häfliger
- Dominik Süsstrunk

2.2 Traktanden

- vServer für Team Server / Version Control
- Bücher für Silverlight (3.0)
    - Bestellen
- FOREX Unklarheiten
    - Meta Traider Daten (Welche Export Daten?)
- Entwurf Projektplan diskutieren

2.3 Beschlüsse

- vServer
    - Direkt bei Verantwortlichen Anfragen
- Bücher / Silverlight infos
    - Franzsika Altdorfer (Bachelorarbeit FS09)
    - Micha Boller & Danny Meier (Bachelorarbeit FS09)
    - LABS auf CD (PhotoBrowser)
    - MVVM von namoSMS (SE2 FS09)
- FOREX
    - DataFeed als Servlet, da Banken wohl so operieren
- Projektplan
    - Anpassungen
        - Arbeitspaket Video hinzufügen
        - Mehr Aufwand für Paket Benutzerhandbuch
- Weiters Vorgehen
    - Anpassen des Projektplans
    - Erstellen der Anforderungen
    - Nächstes Meeting
        - Mittwoch, 7.10.09 17:00
- Weiteres
    - Abgaben Kurzversion
        - Schule
            - Kurzer Text
        - Wiki
            - 5 Seiten
                1. Einleitung (Was ist FOREX? Existierende Lösung, Matrix Lösung)
                2. Silverlight (Was ist SL? Was kann SL? Was kann es nicht?)
                3. Unsere Lösung(Architektur, Netzwerk, Oberfläche
                4. Erfahrungen (Probleme mit SL, Servlets…)

3 Sitzungsprotokoll vom 07.10.09

3.1 Teilnehmer

- Prof. Dr. Markus Stolze
- Daniel Häfliger
- Dominik Süsstrunk

3.2 Traktanden

- Anforderungsspezifikationen
    - Use Cases
- Domainanalyse
    - Domainmodell
    - Contracts
- Server
    - Java
    - SOAP
- Paper Prototype V1
- Daten für Dr. Liver
- weiteres

3.3 Beschlüsse

- Personas, Szenarios, Use Cases an Herrn Liver senden
- Fragen an Herrn Liver
    - OcO/OsO unterschiede der verschiedenen Rows
    - Spot Date bei EBS View (in der alten Applikation pro Tile)
    - Ticket-Grid ausdrucken?
    - Redundanzen in Ticket-Grid nötig? (Buy/Sell für beide Currencies)
    - EBS Views für Kunden/User
    - Namen für Zeilen bei Limited Order (bei uns bisher Rows genannt)
- Domainmodel
    - Row überarbeiten
    - Tile in Domainmodel/SSD
- Sequenzdiagramm
    - Client Start -> Updates
    - Connections to SOAP
- Drucken -> PDF generieren

4 Sitzungsprotokoll vom 23.10.09

4.1 Teilnehmer

- Prof. Dr. Markus Stolze
- Daniel Häfliger
- Dominik Süsstrunk

4.2 Tranktanden

- Besprechung Architekturprototype

4.3 Beschlüsse

- Architekturprototype
    - Anpassen der Struktur
        - GetHistory zu Testbed
        - WCF service nur für User-Verwaltung
    - Überprüfen viewmodels – models
- Evtl frage für Dr. Liver: Wer verwaltet subscriptions? Client oder testbed
- Weiteres Vorgehen:
    - Anpassung Architektur wie oben erwähnt
    - EBS-Chart und Linechart Ansichten erstellen
    - Aufarbeitung Dokumente (SAD, Projektplan, Testdokument)
- Nächster Termin
    - 4.11.09 – Termin mit M. Stolze
    - Woche Danach: Möglichkeit zum Termin mit Dr. Liver

5 Sitzungsprotokoll vom 04.11.09

5.1 Teilnehmer

- Prof. Dr. Markus Stolze
- Daniel Häfliger
- Dominik Süsstrunk

5.2 Tranktanden

- Stand überprüfen
  - EBS Chart
  - Graph
  - SAD

5.3 Beschlüsse

Termin mit Herr Liver festlegen

Ansonsten fortfahren wie gehabt

6 Sitzungsprotokoll vom 19.11.09

6.1 Teilnehmer

- Prof. Dr. Markus Stolze
- Dr. Beat Liver
- Daniel Häfliger
- Dominik Süsstrunk

6.2 Tranktanden

- Prototyp
    - Betrag (USD) pro Konto
    - Detail-Tile Anordnung
    - Anz. Tiles eines Traders
        - CPU/Animationen
    - Differenz Sell/Buy
    - Highlight-Color von Tiles

6.3 Beschlüsse

- UI
    - Key Acceleration
    - Versuch mit Highlight-Color Tiles
    - Graph
        - Kerzendarstellung ab 1min
        - Darstellung von moving average
        - History
        - Graph ohne Animation (prüfen ob nicht zu nervös)
    -
- Stream
    - http Request für Polling
    - Auf Server packen mit Wireshark prüfen / Datenlast berechnen
    - SSL-Sessions analysieren (möglichst nur 1 Session)
    - Binary Encoding
- Auswahl des Zahlungsweges
- Präsentation des Projektes voraussichtlich in der Woche vom 21. Dez
- Demo-Video des Prototyp 2.

7 Sitzungsprotokoll vom 02.12.09

7.1 Teilnehmer

- Prof. Dr. Markus Stolze
- Daniel Häfliger
- Dominik Süsstrunk

7.2 Tranktanden

- Abgaben
    - Abstract
    - Management Summary
    - Poster
    - Hauptabgabe 18.12.09 (Wie)
- Forex UI
    - Schlussspurt
        - Was muss noch zwingend gemacht werden
    - Dokumentation
- Weiteres
    - Gegenleser J. Joller

7.3 Beschlüsse

- Abgaben
    - Abstract
        - Thema aus AVT
    - Management Summary
        - Selber text wie Abstract
    - Poster
        - Stichworte, keine Texte
        - Ziel
    - 1 Dokument mit allem + Abstract
    - Zusätzliche Dokumentation für Herr Stolze
        - Demo Video
        - Wiki Page wie in Sitzung von 23.09 besprochen
- Hilfe -> System durch Tipps at Start zeigen
- Abgabe nächste Woche
    - Draft des 5 Seiten Papers
- Kontakt mit Gegenleser (J. Joller) aufnehmen

8 Sitzungsprotokoll vom 16.12.09

8.1 Teilnehmer

- Prof. Dr. Markus Stolze
- Daniel Häfliger
- Dominik Süsstrunk

8.2 Traktanden

- Management Summary
- Abstract
- Poster
- Server Erhaltung
- Literaturverzeichnis

8.3 Beschlüsse

- Abstract
  - Datenbank zu Testbed?
  - Skalierbarkeit serverseitig (integrieren in folgearbeit evtl…)
  - Dr. Liver Credit Suisse umkehren
- Poster
  - Dr. Liver Credit Suisse umkehren
  - Techniker->Technologien
- Literaturverzeichnis
  - Eigenes, andere bleibenlassen
- Ausdruck
  - 1 x in stolze (schwarz/weiss)
  - 1 x für Dr. Liver (evtl. farbig)
- Abgabe
  - 1 x CD für Sekretariat
  - 1 x CD Dr. Liver
  - 1 x CD Stolze
  - 1x CD self
- Präsentation
  - Kurze Präsentation
    - Was wurde gemacht
    - Was wurde nicht gemacht
    - Zeit Investition
  - Demo

Foreign Exchange UI

Anhang

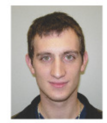
1 Poster

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
INFORMATIK

Semesterarbeit Herbstsemester 09
Software

FOREIGN EXCHANGE USER INTERFACE

User Interface mit Silverlight



Daniel Häfliger



Dominik Süsstrunk

Betreuer
Prof Dr. Markus Stolze

Industriepartner
Dr. Beat Liver (Credit Suisse)

Projektbeschreibung

Ziel
- Machbarkeit Silverlight Applikation im Bereich Forex Trading

Anforderungen Client
- Gute Bedienbarkeit
- Sicherheit
- Einfache Wartbarkeit
- Gute Skalierbarkeit

Anforderungen Testbed
- Erzeugen von Devisenkursen
- Benutzerverwaltung
- Aufträge ausführen





Ergebnis

Testbed
- Kommunikation mit Client über einzelnen WCF Service
- Datenbankzugriff über Linq to SQL
- ASP Session als einzige Sicherheit

User Interface
- Selbsterklärend
- Trennung durch MVVM
- Auslagerung auf GPU

Architektur

Server
- Windows 2003 Server
- SQL Server 2008
- C#

Technologien
- Linq to SQL
- WCF
- MVVM

Prototyp
- Silverlight SL 3
- C# / XAML

Erkenntnisse

Silverlight
- WCF minimal implementiert
- Fehler bei Animationen
- Verbesserung mit SL 4

Performance
- Stark abhängig von Netzwerk
- Animationen verlangsamen UI

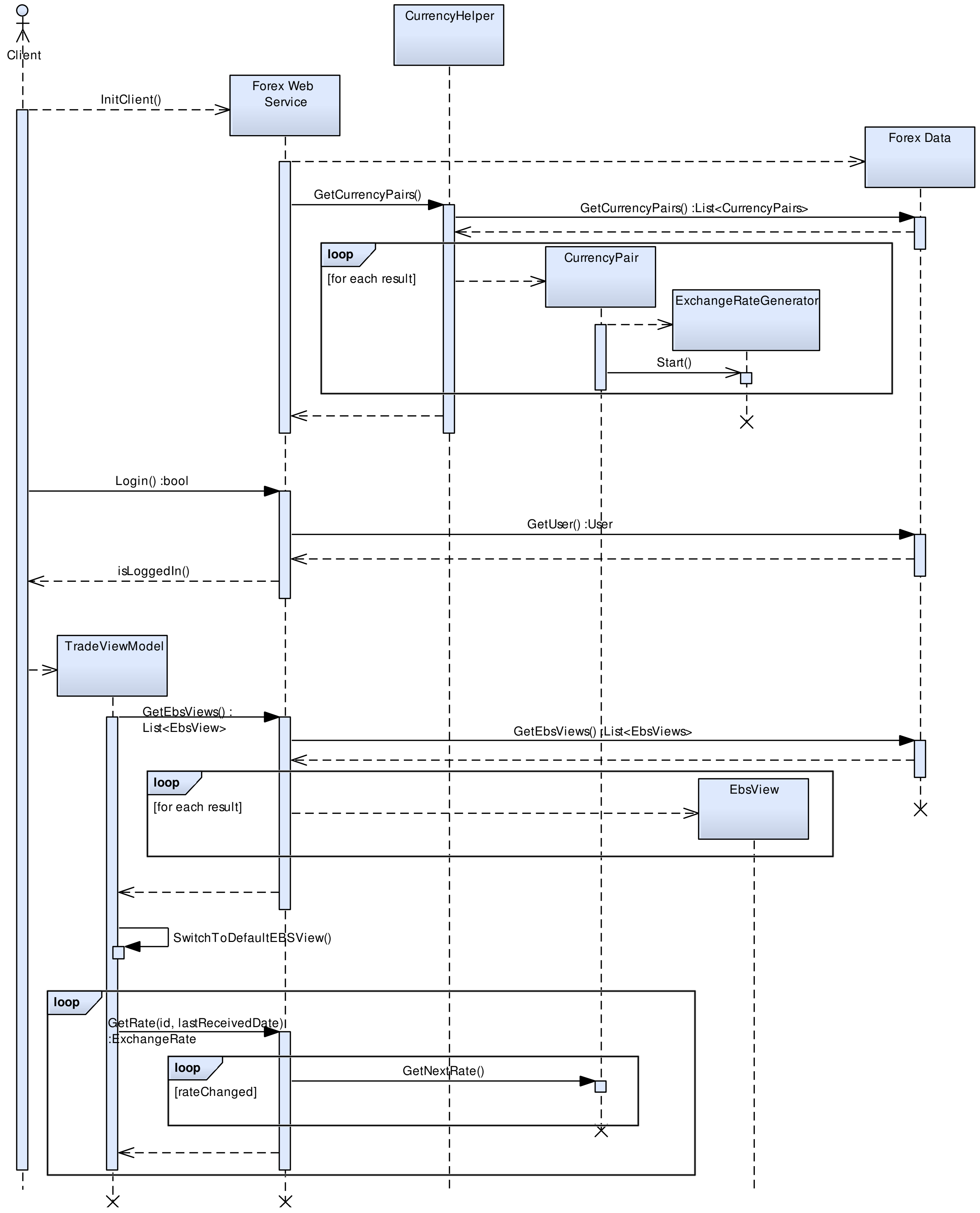
Security
- Möglichkeit über SSL
- Keine Sicherheit bei WCF Service

Foreign Exchange
- Komplizierte Prozesse
- OCO
- OSO



CREDIT SUISSE
Silverlight

2 System Sequenz Diagramm



Währungspaare Laden und für jedes Währungspaar einen ExchangeRateGenerator erstellen, welcher mit Hilfe von Events die Währungspaare aktualisiert.

Benutzer mit gehashtem Passwort (SHA-256) in Datenbank suchen.
Falls der Benutzer gefunden wird, so wird dieser in einer Session gespeichert um die weiteren Anfragen zu authentisieren.

Nach dem Anmelden wird ein TradeViewModel erstellt, dieses lädt alle EbsViews, welche dem Benutzer zugeordnet sind.

Die geladenen EbsViews werden in einer Liste hinzugefügt und diese wird vom Service zurückgegeben.

Sind alle EbsViews geladen, so wird zur Default EbsView gewechselt.

Für alle in der Default EbsView angezeigten Währungspaare werden nun die Raten Abonniert.

Nach jeder Antwort einer Ratenanfrage, wird eine neue Anfrage der betreffenden Rate gesendet.